



**Titre:** Modélisation 3D de vaisseaux sanguins à partir d'images  
Title: médicales : approche uniformisante à l'aide de séquences de filtres  
3D

**Auteur:** Méliissa Côté  
Author:

**Date:** 2006

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Côté, M. (2006). Modélisation 3D de vaisseaux sanguins à partir d'images  
Citation: médicales : approche uniformisante à l'aide de séquences de filtres 3D [Mémoire  
de maîtrise, École Polytechnique de Montréal]. PolyPublie.  
<https://publications.polymtl.ca/7706/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:**  
PolyPublie URL: <https://publications.polymtl.ca/7706/>

**Directeurs de  
recherche:**  
Advisors:

**Programme:** Non spécifié  
Program:

UNIVERSITÉ DE MONTRÉAL

MODÉLISATION 3D DE VAISSEaux SANGUINS À PARTIR  
D'IMAGES MÉDICALES : APPROCHE UNIFORMISANTE  
À L'AIDE DE SÉQUENCES DE FILTRES 3D

MÉLISSA CÔTÉ  
DÉPARTEMENT DE GÉNIE INFORMATIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE INFORMATIQUE)  
AVRIL 2006



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file    Votre référence*

*ISBN: 978-0-494-17937-6*

*Our file    Notre référence*

*ISBN: 978-0-494-17937-6*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MODÉLISATION 3D DE VAISSEAUX SANGUINS À PARTIR  
D'IMAGES MÉDICALES : APPROCHE UNIFORMISANTE  
À L'AIDE DE SÉQUENCES DE FILTRES 3D

présenté par : CÔTÉ Mélissa

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

Mme CHERIET Farida, Ph.D., présidente

M. OZELL Benoît, Ph.D., membre et directeur de recherche

M. MERLO Ettore, Ph.D., membre



## REMERCIEMENTS

Je tiens à remercier les gens ayant contribué à la réalisation de ce mémoire. Je voudrais tout d'abord exprimer ma reconnaissance à mon directeur de recherche, M. Benoît Ozell, pour m'avoir guidée tout au long du parcours, pour m'avoir aidée à développer les habiletés nécessaires à la recherche scientifique, pour m'avoir offert un support financier et pour sa grande disponibilité lors des moments les plus importants. Je remercie particulièrement Marie-Noëlle Lincourt-Éthier, associée de recherche au département de génie mécanique, pour m'avoir fourni les données sources utilisées dans ce projet, de même que de nombreux conseils et renseignements sur le domaine biomédical et sur d'autres sujets. Je veux également exprimer ma gratitude envers le personnel administratif et enseignant du département, pour leur assistance et leur coopération. Je tiens à remercier le Conseil de recherches en sciences naturelles et en génie du Canada pour m'avoir octroyé une bourse d'études supérieures dans le cadre de ce mémoire. Enfin, ce mémoire n'aurait pas été possible sans les encouragements et le soutien formidable de mon conjoint Charles.

## RÉSUMÉ

L'imagerie médicale *in vivo* permet d'obtenir, de façon non intrusive, des séries d'images anatomiques illustrant les différents tissus en trois ou quatre dimensions. Ces images renferment des informations réelles pouvant être traitées numériquement pour obtenir des modèles 3D d'organes afin d'apporter des voies supplémentaires d'étude et d'interprétation. Pour être utilisés avec confiance, ces modèles 3D doivent être fidèles à la réalité. Pourtant, les conditions d'acquisition génèrent du bruit sur les images, compliquant leur utilisation.

Ce mémoire s'inscrit au coeur de cette problématique et traite de la création de surfaces 3D de vaisseaux sanguins à partir d'images médicales 3D produites par résonance magnétique. La revue bibliographique suggère que dans bon nombre d'approches, les auteurs créent un modèle surfacique polygonal en premier lieu (soit directement à partir de l'ensemble des images ou à partir de segments 2D préalablement trouvés image par image) ou après quelques opérations élémentaires sur les images médicales. Les auteurs tentent ensuite d'améliorer la fidélité du modèle reconstruit par des opérations sur les polygones. L'idée maîtresse de ce mémoire est de travailler directement sur les intensités des images, c'est-à-dire conserver le format original « bain de voxels », pour extraire les informations et améliorer la qualité du modèle. Il s'agit d'une approche uniformisante qui intègre certains des avantages des méthodes de segmentation directe et des modèles déformables 3D, tout en permettant de profiter facilement de la cohérence 3D des données sources et de retrouver un modèle géométrique réaliste par des opérations simples. Notre cadre de travail logiciel plante la modélisation 3D par des séquences de filtres unitaires applicables sur les voxels, avant de créer la surface finale par la méthode des cubes baladeurs. La validation est effectuée qualitativement et quantitativement à l'aide de modèles analytiques et réels de la crosse aortique, de l'aorte et des carotides. Les surfaces 3D produites par notre logiciel sont d'allure lisse et globalement en accord avec l'anatomie humaine. Les séquences de filtres ont été

efficaces pour éliminer le bruit présent sur les données sources et pour produire des objets d'intensité uniforme et aux frontières relativement lisses. Les temps d'exécution des séquences sont courts et une implantation sur des processeurs graphiques modernes produirait des temps encore plus rapides, ouvrant la voie à une utilisation en temps réel.

## ABSTRACT

Non-invasive *in vivo* medical imaging enables the creation of series of anatomical images illustrating different bodily tissues in three or four dimensions. These images contain real information which can be digitally processed in order to obtain 3D organ models. As a result, we have new means of study and interpretation. To be relied upon, these models must be exact representations of the real organs. However, in the course of data acquisition, medical images gather noise pollution and using them becomes more complicated.

This thesis addresses the abovementioned problem and deals with the creation of 3D surfaces of blood-vessels from medical images which were produced by magnetic resonance imaging. The literature review suggests that, in several approaches, the authors create a polygonal model of the surface first (either directly from the whole set of images or from 2D segments previously determined from one image at a time) or after a few basic operations on the medical images. Afterwards, the authors try to improve the reconstructed model's quality with operations on the polygons. The preferred approach is to work directly on image intensity values, which means keeping the original "voxel bath" format to extract the information and improve the quality of the model. This is a unifying approach, incorporating some of the benefits from such methods as direct segmentation and 3D deformable models while taking advantage of the original data 3D coherence, resulting in a realistic geometrical model through simple operations. Our framework implements 3D modeling through sequences of unitary filters applicable on the voxels before creating the final surface using the marching cubes method. Qualitative and quantitative validation is done with the help of analytical as well as real models of the aortic arch, the aorta and the carotids. The 3D surfaces produced with our software have a smooth appearance and are on the whole accurate reproductions of the human anatomy. The filter sequences have proven to be effective at eliminating the noise on the original data and at producing objects of uniform intensity with smooth

borders. The sequence execution times are short. Moreover, their implementation in modern graphical processing units could produce even faster times opening up the way to real-time performance.

## TABLE DES MATIÈRES

REMERCIEMENTS .....	iv
RÉSUMÉ .....	v
ABSTRACT .....	vii
TABLE DES MATIÈRES .....	ix
LISTE DES TABLEAUX.....	xii
LISTE DES FIGURES.....	xiii
LISTE DES SIGLES ET ABRÉVIATIONS .....	xviii
LISTE DES ANNEXES.....	xxi
INTRODUCTION .....	1
CHAPITRE 1 : REVUE BIBLIOGRAPHIQUE .....	7
1.1 Introduction .....	7
1.2 Segmentation 2D.....	9
1.2.1 Détection de contours.....	10
1.2.2 Modèles déformables .....	11
1.2.3 Croissance de région .....	18
1.3 Reconstruction 3D à partir de segments 2D.....	20
1.3.1 Approximation / interpolation par des splines .....	21
1.3.2 Cubes baladeurs .....	22
1.3.3 Fragmentation directe .....	25
1.3.4 Flux optique .....	26
1.4 Segmentation 3D.....	28
1.4.1 Détection de contours.....	28
1.4.2 Modèles déformables .....	29
1.4.3 Particules dynamiques.....	38

1.4.4 Propagation d'ondes et traçage arrière.....	39
1.5 Lissage de surfaces.....	41
1.6 Objectifs et hypothèse du mémoire.....	43
CHAPITRE 2 : TRAITEMENTS SUR LES DONNÉES SOURCES.....	48
2.1 Augmentation du contraste .....	50
2.2 Filtre de moyenne pondérée .....	52
2.3 Filtre médian .....	54
2.4 Opération morphologique .....	56
2.4.1 Dilatation.....	56
2.4.2 Érosion .....	58
2.4.3 Ouverture / fermeture.....	60
2.5 Croissance de région .....	61
CHAPITRE 3 : MISE EN OEUVRE LOGICIELLE.....	69
3.1 Architecture logicielle.....	69
3.1.1 Sous-système « Application ».....	72
3.1.2 Sous-système « Interface Graphique ».....	73
3.1.3 Sous-systèmes « Formats Entrée » / « Formats Sortie ».....	74
3.1.4 Sous-système « Données ».....	74
3.1.5 Sous-systèmes « Usines Traitements » / « Traitements ».....	75
3.1.6 Sous-système « Créations Surfaces ».....	76
3.1.7 Sous-systèmes « Surfaces » / « Éléments » .....	77
3.1.8 Sous-système « Métriques ».....	77
3.2 Aspects comportementaux .....	78
3.3 Autres considérations .....	81
CHAPITRE 4 : RÉSULTATS .....	83
4.1 Série crosse aortique et carotides .....	83
4.2 Série aorte.....	89

4.3 Données analytiques.....	91
4.4 Temps d'exécution.....	96
4.5 Exportation des surfaces en formats « VU » et « YAMS » .....	97
CHAPITRE 5 : DISCUSSION.....	100
5.1 Discussion sur la série crosse aortique et carotides .....	100
5.1.1 Séquences de traitements et surfaces résultantes .....	100
5.1.2 Validation.....	105
5.2 Discussion sur la série aorte.....	112
5.2.1 Séquences de traitements et surfaces résultantes .....	112
5.2.2 Validation.....	114
5.3 Discussion sur les données analytiques.....	115
5.3.1 Séquences de traitements et surfaces résultantes .....	115
5.3.2 Validation.....	117
5.4 Discussion sur les temps d'exécution et la complexité.....	118
5.5 Discussion sur la méthodologie, l'hypothèse et les objectifs.....	120
CONCLUSION.....	124
BIBLIOGRAPHIE .....	128
ANNEXES .....	139



## LISTE DES TABLEAUX

Tableau 2.1 : Caractéristiques principales des traitements retenus.....	68
Tableau 4.1 : Informations relatives à la surface pour la série crosse aortique et carotides. .....	89
Tableau 4.2 : Informations relatives à la surface pour la série aorte.....	91
Tableau 4.3 : Informations relatives à la surface pour la série données analytiques. ....	95
Tableau 4.4 : Erreurs sur l'aire et le volume pour la série données analytiques.....	96
Tableau 4.5 : Temps moyens d'exécution des algorithmes des traitements et écarts types pour les trois séries de données (temps pour les séries complètes et temps par Mvoxels (1 000 000 voxels)). ....	97
Tableau C.1 : Commandes et paramètres pour le format de fichier « TRM ». ....	163

## LISTE DES FIGURES

Figure 0.1 : Exemples d'images médicales (IRM) de la crosse aortique et des carotides tirées d'une série de 140 images. ....	4
Figure 1.1 : Application du filtre de Canny. ....	10
Figure 1.2 : Évolution d'un contour actif classique (courbe fermée) pour la segmentation d'un anneau. ....	13
Figure 1.3 : Ensemble de niveaux : courbe $C$ (dans $\mathcal{R}^2$ ) correspondant à la valeur nulle d'une autre fonction $u$ . ....	15
Figure 1.4 : Adaptation topologique de l'ensemble de niveaux.....	15
Figure 1.5 : Segmentation avec un contour actif discriminant statistique. ....	18
Figure 1.6 : Problème de segmentation avec le contour actif discriminant statistique. ....	18
Figure 1.7 : Voisins immédiats d'un voxel dans l'algorithme de croissance de région. ....	19
Figure 1.8 : Reconstruction de la surface paramétrique d'une artère avec des splines par interpolation en a) et par approximation précédée d'une augmentation des données en b).....	21
Figure 1.9 : Algorithme des cubes baladeurs : exemple de configuration possible de l'intersection de la surface de l'objet avec le cube logique. ....	23
Figure 1.10 : Reconstruction d'un vaisseau sanguin a) par l'algorithme des cubes baladeurs, b) par l'algorithme de préservation du volume.....	25
Figure 1.11 : Fragmentation directe des voxels frontaliers.....	25
Figure 1.12 : Reconstruction de la carotide avec la fragmentation directe.....	26
Figure 1.13 : Éléments utilisés dans l'algorithme de reconstruction de surfaces polygonales basé sur le flux optique. ....	27
Figure 1.14 : Évolution d'un ballon 3D dans le processus de segmentation de la bifurcation des carotides. ....	31
Figure 1.15 : Modèle déformable cylindrique, représenté par un axe central et une paroi. ....	33
Figure 1.16 : Méthode à bande étroite pour les ensembles de niveaux.....	35

Figure 1.17 : Évolution des bulles 3D dans le processus de segmentation d'une haltère.	36
Figure 1.18 : Segmentation 3D avec des bulles.	37
Figure 1.19 : Segmentation 3D de données provenant d'angiographie par résonance magnétique par propagation d'une onde et traçage arrière.	40
Figure 2.1 : Exemples de noyaux pour l'application de filtres sur des données tridimensionnelles.	49
Figure 2.2 : Application du filtre d'augmentation du contraste avec des seuils minimal et maximal de 0,3 et 0,4.	52
Figure 2.3 : Noyau du filtre de moyenne pondérée.	53
Figure 2.4 : Application du filtre de moyenne pondérée, avec des poids $W_i$ et $W_j$ de 2/34, $W_k$ de 3/34 et $W_{autres}$ de 1/34.	54
Figure 2.5 : Application du filtre médian.	55
Figure 2.6 : Dilatation binaire avec un élément structurant carré de 3 x 3 pixels.	57
Figure 2.7 : Application du filtre de dilatation fonctionnelle.	58
Figure 2.8 : Érosion binaire avec un élément structurant carré de 3 x 3 pixels.	59
Figure 2.9 : Application du filtre d'érosion fonctionnelle.	60
Figure 2.10 : Émulation de la segmentation d'une image binaire par un modèle déformable comme le ballon.	63
Figure 2.11 : Voisinage et position des pixels R et V pour la croissance de région.	64
Figure 2.12 : Croissance de région avec pixel V d'intensité acceptable.	64
Figure 2.13 : Croissance de région avec pixel V d'intensité non acceptable.	65
Figure 2.14 : Croissance de région : illustration de la fenêtre 3 x 3 x 3 centrée sur le voxel immédiatement voisin en traitement (V).	66
Figure 2.15 : Application de l'algorithme de croissance de région.	67
Figure 3.1 : Flux global d'opérations de la catégorie particulière de logiciels visée.	70
Figure 3.2 : Organisation des sous-systèmes du programme.	72
Figure 3.3 : Visualisation d'images médicales selon trois coupes : a) axiale, b) coronale et c) sagittale.	80

Figure 4.1 : Surfaces 3D pour la série crosse aortique et carotides. ....	84
Figure 4.2 : Séquence de traitements pour la série crosse aortique et carotides. ....	85
Figure 4.3 : Surfaces 3D pour la série crosse aortique et carotides, cas # 2. ....	86
Figure 4.4 : Séquence de traitements pour la série crosse aortique et carotides, cas # 2. ....	87
Figure 4.5 : Surfaces 3D pour la série crosse aortique et carotides tronquée, cas # 3. ....	87
Figure 4.6 : Séquence de traitements pour la série crosse aortique et carotides, cas # 3. ....	88
Figure 4.7 : Surfaces 3D pour la série aorte. ....	90
Figure 4.8 : Séquence de traitements pour la série aorte. ....	90
Figure 4.9 : Dimensionnement des données analytiques. ....	92
Figure 4.10 : Surfaces 3D pour les données analytiques, sans traitement. ....	92
Figure 4.11 : Surfaces 3D pour les données analytiques, après le filtre de moyenne pondérée. ....	93
Figure 4.12 : Application de bruit sur les données analytiques. ....	94
Figure 4.13 : Surfaces 3D pour les données analytiques bruitées. ....	94
Figure 4.14 : Séquence de traitements pour la série données analytiques bruitées. ....	95
Figure 4.15 : Exportation de la surface issue de la série crosse aortique et carotides en formats « VU » (a) et « YAMS » (b). ....	98
Figure 5.1 : Effets de la séquence de traitements sur la série crosse aortique et carotides (cas #1), étape par étape pour une image en coupe axiale (image numéro 18 sur 140). ....	101
Figure 5.2 : Extrémité de la crosse aortique dans les données sources, mal définie en a) et bien définie en b). ....	105
Figure 5.3 : Section des images originales (colonne de gauche) versus images segmentées (colonne de droite) pour le cas #1 de la série crosse aortique et carotides. ....	106
Figure 5.4 : Sections des images originales (colonne de gauche) versus images segmentées (colonne de droite) pour le cas #2 de la série crosse aortique et carotides. ....	108

Figure 5.5 : Géométrie <i>in vivo</i> (images de gauche) versus surface reconstruite par notre logiciel (images de droite) pour le cas #1 de la série crosse aortique et carotides. ....	109
Figure 5.6 : Segmentation manuelle (images de gauche) versus surface reconstruite par notre logiciel (images de droite) pour le cas #1 de la série crosse aortique et carotides. ....	110
Figure 5.7 : Effets de la séquence de traitements sur la série aorte étape par étape pour une section de l'image 5 en coupe axiale.....	113
Figure 5.8 : Sections des images originales (rangée du haut) versus images segmentées (rangée du bas) pour la série aorte. ....	114
Figure 5.9 : Effets de la séquence de traitements sur les données analytiques bruitées, étape par étape, pour une section d'image en coupe sagittale (image numéro 78). ....	116
Figure 5.10 : Sections de l'image numéro 78 originale (a) versus image bruitée (b) versus image segmentée (c) pour la série données analytiques en coupe sagittale. ....	117
Figure A.1 : Images médicales de la série crosse aortique et carotides. ....	139
Figure A.2 : Images médicales de la série aorte.....	151
Figure B.1 : Algorithme d'augmentation du contraste.....	152
Figure B.2 : Algorithme du filtre de moyenne pondérée. ....	153
Figure B.3 : Algorithme du filtre médian. ....	154
Figure B.4 : Algorithme de dilatation et d'érosion fonctionnelles. ....	155
Figure B.5 : Algorithme de croissance de région.....	158
Figure C.1 : Format de fichier « VOL ». ....	161
Figure C.2 : Format de fichier « TRM ». ....	162
Figure C.3 : Format de fichier « VU ». ....	164
Figure C.4 : Format de fichier « YAMS ». ....	165
Figure D.1 : Diagramme du sous-système « Application ». ....	167
Figure D.2 : Diagramme du sous-système « Interface Graphique ». ....	168

Figure D.3 : Diagramme du sous-système « Formats Entrée ».....	168
Figure D.4 : Diagramme du sous-système « Formats Sortie ». ....	169
Figure D.5 : Diagramme du sous-système « Données ». ....	169
Figure D.6 : Diagramme des sous-systèmes « Usines Traitements » et « Traitements ». ....	170
Figure D.7 : Diagramme du sous-système « Créations Surfaces ». ....	171
Figure D.8 : Diagramme du sous-système « Surfaces ». ....	171
Figure D.9 : Diagramme du sous-système « Éléments ». ....	172
Figure D.10 : Diagramme du sous-système « Métriques ». ....	172
Figure E.1 : Syntaxe à suivre lors du lancement d'ASSIMMO3D en mode console. ....	173
Figure E.2 : Diagramme de séquence UML pour le mode console. ....	174
Figure E.3 : Fenêtre de contrôle de l'affichage d'ASSIMMO3D en mode interactif. ....	175
Figure E.4 : Interface usager graphique pour le choix des paramètres pour les traitements. ....	176

## LISTE DES SIGLES ET ABRÉVIATIONS

2D / 3D / 4D	deux, trois ou quatre dimensions
$a$	rayon de courbure d'un tore
$Aire_{reconstruite}$	aire de la surface reconstruite par le logiciel
$Aire_{réelle}$	aire de la surface d'un modèle mathématique
ARM	angiographie par résonance magnétique
ASSIMMO3D	Architecte de Segmentation de Séries d'Images Médicales et de Modélisation d'Organes 3D
$b$	bruit de type gaussien
$C$	équation de l'ensemble de niveau zéro (modèle déformable)
$D_t$	temps entre deux séries de données
$D_x$	espacement entre le centre de deux voxels voisins en $x$
$D_y$	espacement entre le centre de deux voxels voisins en $y$
$D_z$	espacement entre le centre de deux voxels voisins en $z$
DICOM	<i>Digital Imaging and Communications in Medicine</i>
$E_{in}$	énergie interne de déformation pour le contour actif
$E_{im}$	énergie associée au contenu de l'image pour le contour actif
$E_{con}$	énergie associée aux contraintes pour le contour actif
$E(v(s))$	énergie du contour actif
$Erreur_{aire}$	erreur en pourcentage sur l'aire de la surface
$Erreur_{volume}$	erreur en pourcentage sur le volume interne de la surface
$F$	vitesse du modèle déformable représenté avec les ensembles de niveaux
GLUT	<i>OpenGL Utility Toolkit</i>
GPU	<i>Graphical Processing Unit</i>
$h(l,m,n)$	valeur du filtre correspondant aux indices $l,m,n$
$I_{finale}(i,j,k)$	intensité après traitement du voxel situé aux indices $i,j,k$
$I_{initiale}(i,j,k)$	intensité avant traitement du voxel situé aux indices $i,j,k$

$I_{\max}$	valeur maximale possible d'un voxel
IRM	imagerie par résonance magnétique
K	côté de la fenêtre cubique d'un filtre
L	longueur d'un cylindre
N	nombre de voxels dans une série de données
$N_i$	taille des données (nombre de voxels) en i
$N_j$	taille des données (nombre de voxels) en j
$N_k$	taille des données (nombre de voxels) en k
$n_M$	nombre de voxels ayant comme intensité $i_M$
NbVoxelsFenêtreOK	nombre de voxels d'une fenêtre 3 x 3 x 3 ayant une intensité comprise dans la bonne plage de valeurs
NbVoxelsPlanOK	nombre de voxels d'un plan 3 x 3 ayant une intensité comprise dans la bonne plage
NbVoxelsVImmédiatsOK	nombre de voxels immédiatement voisins (partageant une face) ayant une intensité comprise dans la bonne plage
$p(i_M)$	valeur de l'histogramme pour l'intensité $i_M$
r	rayon d'un cercle, d'un cylindre ou d'un tore
R	voxel (ou pixel) d'où provient la croissance de région
Seuil <sub>min</sub>	seuil minimal pour un traitement
Seuil <sub>max</sub>	seuil maximal pour un traitement
STL	<i>Standard Template Library</i>
u	fonction dans un espace à dimensions plus élevées utilisée pour les ensembles de niveaux
UML	<i>Unified Modeling Language</i>
$v(s)$	équation du contour actif en fonction du paramètre s
V	voxel immédiatement voisin (partageant une face) d'un autre voxel, ou pixel immédiatement voisin (partageant un côté) d'un autre pixel
VIMoins1	voxel immédiatement voisin en i-i



VIPlus1	voxel immédiatement voisin en $i+i$
VJMoins1	voxel immédiatement voisin en $j-i$
VJPlus1	voxel immédiatement voisin en $j+i$
VKMoins1	voxel immédiatement voisin en $k-i$
VKPlus1	voxel immédiatement voisin en $k+i$
Volume <sub>reconstruit</sub>	volume interne de la surface reconstruite par le logiciel
Volume <sub>réel</sub>	volume interne de la surface d'un modèle mathématique
voxel	élément de volume
$W_i$	poids des voxels immédiatement voisins en $i$
$W_j$	poids des voxels immédiatement voisins en $j$
$W_k$	poids des voxels immédiatement voisins en $k$
$W_{\text{autres}}$	poids des voxels voisins partageant seulement un coin avec le voxel central

## **LISTE DES ANNEXES**

ANNEXE A : DONNÉES SOURCES MÉDICALES.....	139
ANNEXE B : DESCRIPTION DES ALGORITHMES DES FILTRES PROPOSÉS ..	152
ANNEXE C : FORMAT DES FICHIERS UTILISÉS .....	160
ANNEXE D : DIAGRAMMES DES SOUS-SYSTÈMES .....	167
ANNEXE E : FONCTIONNEMENT ET EXÉCUTION D'ASSIMMO3D .....	173

## INTRODUCTION

Le développement des méthodes d'acquisition de données tridimensionnelles dans les dernières décennies a ouvert la voie à une panoplie de nouvelles analyses possibles dans diverses disciplines scientifiques, particulièrement dans le domaine biomédical. Avec les techniques d'imagerie médicale *in vivo*, il est possible d'obtenir, de façon non intrusive, des séries d'images anatomiques illustrant les différents tissus organiques en trois ou même en quatre dimensions. Ces volumes de données médicales renferment des informations réelles qui peuvent être traitées numériquement afin d'apporter des voies supplémentaires d'étude et d'interprétation sur les tissus organiques.

La modélisation tridimensionnelle d'un organe est un exemple répandu d'utilisation d'images médicales. Les applications d'une telle représentation fidèle des organes *in situ* sont nombreuses (Shahidi, Tombropoulos et Grzeszczuk 1998, Nikolaidis et Pitas 2001) :

- Diagnostic : La visualisation tridimensionnelle des données médicales, alliée à d'autres mesures et observations, peut aider le clinicien dans sa pose de diagnostic sur des tumeurs ou des lésions. La reconstruction 3D offre de nouvelles vues sur les organes, impossibles directement par les images médicales. Toutefois, le manque de standardisation dans le processus de reconstruction des modèles est l'un des facteurs expliquant le lent progrès de l'utilisation de cette technique dans la pratique clinique moderne.
- Planification de chirurgies : L'utilisation de modèles tridimensionnels peut faciliter la planification préopératoire et minimiser l'exploration lors de chirurgies, par exemple dans la planification du meilleur chemin pour l'insertion de l'instrumentation.
- Évaluation postopératoire : Des modèles 3D de l'organe d'intérêt reconstruits à partir d'images médicales prises avant et après une opération chirurgicale peuvent être examinés dans le but d'évaluer la réussite de l'opération.

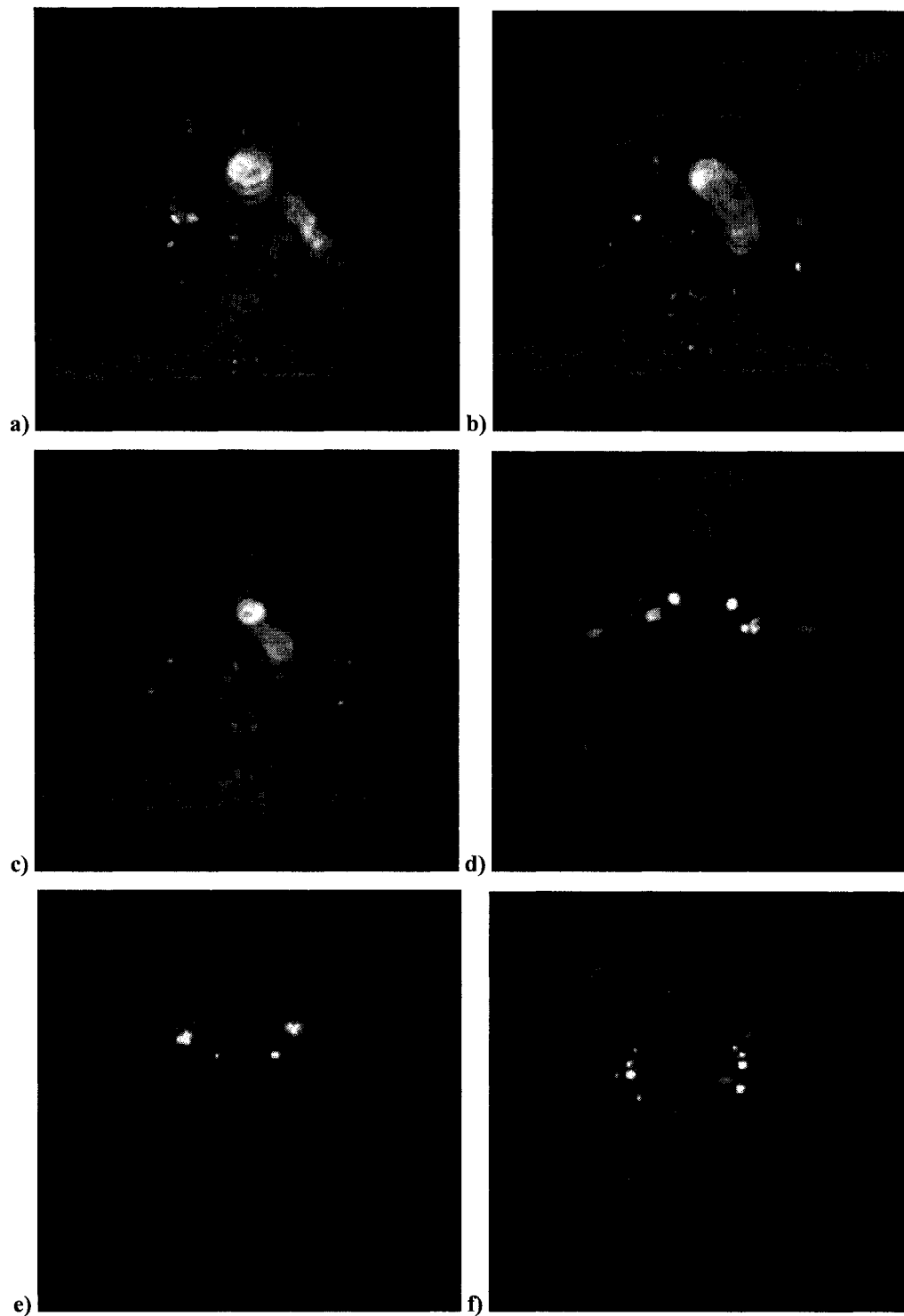
- Robotique chirurgicale et télémedecine : Les modèles informatisés d'organes peuvent servir à guider des robots dans l'accomplissement de tâches chirurgicales. Combinés avec des accessoires générant un retour de force, les modèles peuvent être utilisés dans des systèmes de téléprésence, dans lesquels un chirurgien peut surveiller et performer certaines opérations à distance.
- Entraînement de personnel médical : Dans l'enseignement de l'anatomie, des organes modélisés de façon réaliste à partir d'images médicales pourraient constituer une alternative à l'utilisation de cadavres. Lorsque agencés à des modèles biomécaniques et physiologiques simulant les effets d'opérations sur le corps humain et à des techniques de réalité virtuelle, ces reconstructions d'organes peuvent faire partie de l'entraînement du personnel médical pour certains types d'interventions, telle une chirurgie orthopédique.
- Simulations numériques : Un autre vaste domaine d'application de la modélisation 3D d'organes à partir d'images médicales est la simulation numérique d'écoulement de fluide. À titre d'exemple, à partir de modèles du système vasculaire, il est possible de simuler l'écoulement sanguin, ce qui permet la visualisation du comportement du fluide et des différentes forces qui en résultent. Cela fournit une méthode supplémentaire (à l'analyse directe des images médicales) pour la détection de sites propices aux pathologies, et facilite la calibration d'un appareil médical pour un patient, tel une pompe cardiaque.

L'imagerie par résonance magnétique (IRM) constitue l'une des techniques d'imagerie médicale les plus couramment utilisées pour la reconstruction de modèles 3D anatomiques (Shahidi, Tombropoulos et Grzeszczuk 1998). Elle est basée sur les propriétés magnétiques détectables de certains noyaux atomiques, le plus souvent des atomes d'hydrogène. Les images sont reconstituées à partir des temps de relaxation des atomes préalablement excités par un champ magnétique à une fréquence spécifique (Coatrieux et Roux 1997). Puisque la densité des atomes d'hydrogène varie d'un tissu à l'autre, les divers organes répondent de façon différente. Par exemple, les os et l'air

produisent des pixels d'intensités foncées car ils contiennent des quantités minimales d'hydrogène, alors que le gras et l'eau, de par leur composition élevée en hydrogène, donnent lieu à des pixels d'intensités claires (Shahidi, Tombropoulos et Grzeszczuk 1998).

L'acquisition des images se fait plan par plan, dans le plan XY de l'objet. Lorsqu'une image a été retrouvée, le détecteur de résonance magnétique se déplace dans le plan Z pour acquérir l'image suivante. Ce processus produit donc une pile d'images en deux dimensions, qui peuvent être vues comme des coupes transversales. Le fait d'empiler les images 2D mène à la troisième dimension et donc à des données volumétriques. Les quantités de données produites par une session d'imagerie sont substantielles : généralement, les images produites par IRM ont une résolution de 256 x 256 pixels, et une série peut contenir un nombre d'images dépassant la centaine. Les images représentent des dizaines de mégaoctets qui, grâce aux avancements technologiques de l'informatique, peuvent être traités sans problème par la plupart des ordinateurs personnels. Cela contribue à la popularité de la manipulation d'images médicales telle que la modélisation géométrique tridimensionnelle d'organes.

Pour être utilisés avec confiance, les modèles géométriques 3D des objets se doivent d'être des représentations fidèles de la réalité, particulièrement dans le domaine médical. Pourtant, les conditions réelles d'acquisition ajoutent des artéfacts sur les images. La déformation des organes dans le temps, due entre autres à la respiration, et l'effet de volume partiel causé par la présence possible de plusieurs tissus différents dans l'épaisseur d'une coupe, de même que les étapes de reconstruction du modèle à partir des images introduisent des incertitudes sur le résultat final. La figure 0.1 donne des exemples d'images médicales acquises par IRM qui seront utilisées dans ce mémoire. Les séries complètes sont présentées à l'annexe A.



**Figure 0.1 : Exemples d'images médicales (IRM) de la crosse aortique et des carotides tirées d'une série de 140 images. En a), image numéro 1; en b), image numéro 12; en c), image numéro 19; en d), image numéro 46; en e), image numéro 97; en f), image numéro 126.**

Ce mémoire s'inscrit au coeur de cette problématique entourant la modélisation 3D à partir d'images médicales. Il traite de la reconstruction de modèles tridimensionnels d'objets à partir de données tridimensionnelles, plus spécifiquement de la création de surfaces 3D de vaisseaux sanguins à partir d'images produites par IRM, en portant une attention particulière sur l'aspect logiciel.

Le premier chapitre est consacré à la revue de l'état des avancements et travaux dans le domaine de la modélisation géométrique tridimensionnelle d'organes à partir d'images médicales. Les différentes méthodes présentées sont divisées en deux grandes approches : 1) celles qui extraient les données sur l'organe en deux dimensions, image par image, puis procèdent à une reconstitution (assemblage) en un modèle à trois dimensions; 2) celles qui extraient les données sur l'organe directement en trois dimensions à partir du volume constitué par la série d'images. Le chapitre est clos par la définition, à la lumière de la revue bibliographique, de l'objectif principal, de l'hypothèse de travail et des objectifs spécifiques de ce mémoire.

Le second chapitre porte sur la description de la méthodologie employée pour répondre à l'objectif du projet. Les traitements appliqués aux données sources dans le but de retrouver l'organe d'intérêt sont expliqués dans un vocabulaire uniformisé, dans le but de faciliter la correspondance entre les traitements.

Le chapitre suivant aborde la mise en oeuvre logicielle du projet à travers l'architecture du système et son fonctionnement. Les différents aspects de la conception du logiciel, tels que sa philosophie générale, l'organisation des classes et les choix d'utilisation de bibliothèques logicielles, de même que l'exécution du logiciel y sont discutés.

Le chapitre 4 présente les résultats du système de façon qualitative et quantitative. L'aspect qualitatif est illustré par des images, alors que l'aspect quantitatif porte sur la validation à l'aide de quelques métriques : volume global, aire de la surface. Plusieurs

sources de données sont considérées : images médicales (IRM de la crosse aortique et des carotides, IRM de l'aorte) et données provenant d'un modèle analytique. Ce chapitre propose également des séquences de traitements pour quelques séries d'images typiques.

Le cinquième et dernier chapitre s'intéresse à l'analyse des résultats : les surfaces produites et les séquences de traitements utilisées sont discutées et validées. La validité de la méthode de reconstruction de modèle 3D à partir de données 3D proposée et son aspect uniformisant sont examinés. Le chapitre est conclu par un retour sur les objectifs et l'hypothèse du mémoire tels que définis au premier chapitre.



## CHAPITRE 1 : REVUE BIBLIOGRAPHIQUE

### 1.1 Introduction

Le problème de modélisation géométrique tridimensionnelle d'organes à partir d'images médicales a été traité par la communauté scientifique de deux manières différentes, selon le nombre de dimensions considérées. Dans la première méthode, les données médicales sont extraites en deux dimensions et servent par la suite à créer le modèle en trois dimensions, alors que dans la seconde méthode, les données médicales sont extraites directement en trois dimensions.

La première approche décompose donc la modélisation géométrique de l'organe étudié en deux grandes étapes :

- 1) les sections transversales, c'est-à-dire les images 2D produites par le scanneur, sont d'abord segmentées séparément et généralement indépendamment (il existe des méthodes à 2½D, pour lesquelles la segmentation se fait image par image mais en tenant compte des données des images précédente et suivante);
- 2) elles sont ensuite assemblées en un modèle 3D par un processus de reconstruction.

Les techniques de segmentation utilisées ont pour la plupart été développées dans le domaine de la vision par ordinateur. Dans les méthodes de segmentation répandues, mentionnons le seuillage basé sur l'intensité, la détection des contours, les modèles déformables et les algorithmes de croissance de région. Dans les techniques de reconstruction, on retrouve, entre autres, l'approximation ou l'interpolation par des splines, l'algorithme des cubes baladeurs (*marching cubes*), la fragmentation directe des éléments de volume (voxels) situés aux frontières de l'objet à reconstruire, ainsi que la reconstruction par flux optique.

La seconde grande approche comprend une seule étape, qui consiste à faire une segmentation tridimensionnelle, produisant directement un modèle 3D. La segmentation se fait en utilisant l'ensemble des images, qui forment un volume de données. Les techniques appartenant à cette idéologie s'inspirent dans la plupart des cas des versions 2D, comme par exemple les modèles déformables tridimensionnels tel que le ballon dynamique.

L'extraction des données en deux dimensions est généralement plus simple en termes de calculs que celle en trois dimensions, mais elle ne tient pas compte de la corrélation entre les images, sauf dans le cas des méthodes 2½D qui tirent profit de ces informations supplémentaires de façon limitée. Les organes comportant des embranchements, par exemple les bifurcations du système vasculaire, présentent des difficultés particulières pour la reconstruction à partir des segments 2D lorsqu'un seul organe sur une image apparaît comme deux objets différents sur l'image suivante.

Dans le contexte de la modélisation géométrique d'artères pour des fins médicales, le modèle reconstruit par les méthodes mentionnées précédemment nécessite dans la plupart des cas des opérations visant à adoucir sa surface. Le « lissage de surfaces » est un terme couramment utilisé pour exprimer ce genre d'opérations. L'idée générale est de rendre la position des sommets des polygones formant la surface plus cohérente par rapport à leur voisinage, par exemple en limitant les variations brusques indésirables.

Nous explorerons plus en profondeur quelques-unes des approches de segmentation 2D, reconstruction 3D à partir de segments 2D et segmentation 3D, et introduirons brièvement le lissage de surfaces dans les sections suivantes.

## 1.2 Segmentation 2D

La segmentation dans le domaine du traitement d'images consiste à partitionner l'image en créant des frontières entre les régions présentant des caractéristiques différentes selon un critère spécifique. Dans le cas du présent projet, la segmentation des images médicales signifie la délimitation de l'organe étudié. Les appareils médicaux de résonance magnétique produisent des images sur lesquelles les tissus sont représentés par des intensités différentes. Le critère d'intensité semble a priori être suffisant pour effectuer la segmentation par un seuillage. Toutefois, plusieurs causes de dégradation contribuent à ajouter du bruit, des distorsions ou un aspect flou aux images, ayant pour effet de faire varier de façon appréciable l'intensité d'un même tissu. Ces causes incluent la respiration et les mouvements du patient lorsque l'acquisition des images durent plusieurs minutes (Coatrieux et Roux 1997), les processus d'acquisition et de numérisation (la faiblesse du signal émis par les protons, la taille du voxel (Herment, Pellot et Giovannelli 1997)), l'interférence électromagnétique entre l'instrumentation et d'autres appareils médicaux, etc. Cela rend la tâche de segmentation difficile. Des opérations de prétraitement sont souvent appliquées sur les images afin de faciliter la segmentation. On peut penser, par exemple, à des filtres gaussiens pour minimiser le bruit ou éliminer un niveau de détails inutile, ou encore à des opérateurs d'augmentation de contraste pour accentuer la différence entre les régions. Toutefois, ces opérations ne sont pas prescrites pour tous les cas et la segmentation des images médicales n'en demeure pas moins un important défi.

Plusieurs techniques ont été développées afin de relever le défi posé par les images médicales. Duncan et Ayache (2000) ont fait la revue des progrès dans le domaine de l'analyse d'images médicales et dénotent bien les défis qui s'y rattachent.

Les sections suivantes présentent un portrait des méthodes de segmentation en deux dimensions qui entrent dans la première approche de modélisation, celle en deux étapes :

la détection de contours, les modèles déformables (le contour actif, l'ensemble de niveaux, le modèle géodésique et le modèle probabiliste / statistique) et la croissance de région.

### 1.2.1 Détection de contours

Il existe des opérateurs de bas niveau qui, appliqués sur une image, relèvent à différentes échelles les contours, c'est-à-dire les changements d'intensité. Le filtre de Canny (voir Bose 2004), en est un exemple : il considère les passages par zéro de la deuxième dérivée de l'intensité de l'image, dans la direction du gradient. La figure 1.1 montre l'application du filtre de Canny à différentes échelles sur une image produite par résonance magnétique.

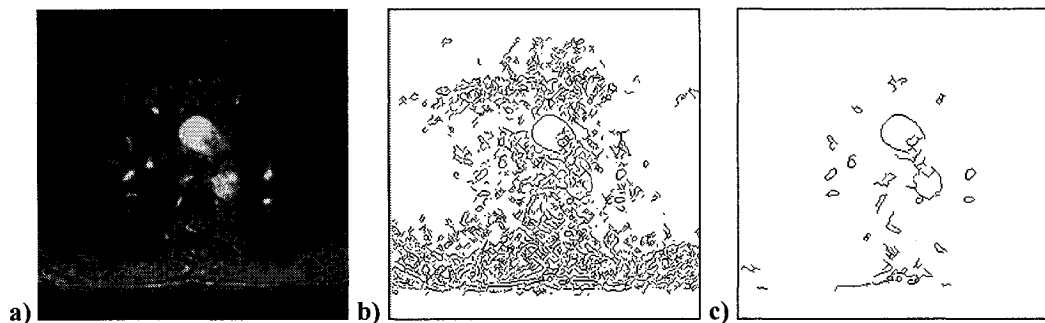


Figure 1.1 : Application du filtre de Canny. Image originale en a) et résultat à deux échelles différentes en b) et c).

De tels opérateurs sont limités dans le domaine de l'imagerie médicale à cause de la présence de régions dégradées. Ils produisent rarement des contours fermés, généralement des résultats erronés et ne peuvent gérer les absences locales de données. Le choix de l'échelle pose également une difficulté.

Kennedy, Filipek et Caviness Jr. (1989) ont proposé une façon d'améliorer l'utilisation de ces opérateurs appliqués à l'imagerie par résonance magnétique en les employant comme approximation de base. Ce plan d'essai initial est par la suite modifié à l'aide

d'une comparaison avec l'opérateur Sobel (voir Russ 1999), une mesure statistique locale de probabilité d'arête. Cette méthode semi-automatique fonctionne suffisamment bien pour permettre l'identification de tissus, mais est très dépendante de la qualité initiale des images médicales.

### **1.2.2 Modèles déformables**

Une façon de pallier aux problèmes des opérateurs locaux de détection de contour est d'incorporer des informations des images à une échelle plus globale. Les modèles déformables exploitent les contraintes provenant des images et peuvent tenir compte des connaissances a priori sur les structures présentes. Ils sont considérés comme des corps élastiques qui répondent à des forces et des contraintes.

Des fonctions d'énergie sont appliquées au modèle qui se déforme sous leur effet. Les déformations sont occasionnées par deux types de forces :

- internes, dues à des contraintes de tension et de rigidité du modèle;
- externes, reliées aux caractéristiques des images.

Afin de tirer profit de l'information connue a priori sur les structures à segmenter, comme l'allure générale, l'orientation et l'emplacement, un troisième type de force peut être ajouté : des contraintes de localisation de forme et de taille spécifiées par l'utilisateur.

Un intéressant survol de l'utilisation des modèles déformables dans l'analyse d'images médicales, bien que relativement vieux, est présenté dans l'article de McInerney et Terzopoulos (1996).

Les prochaines sections présentent quelques modèles déformables qui ont été utilisés pour la segmentation d'images médicales : le contour actif, l'ensemble de niveaux, le modèle géodésique ainsi que le modèle probabiliste / statistique.

### 1.2.2.1 Contour actif

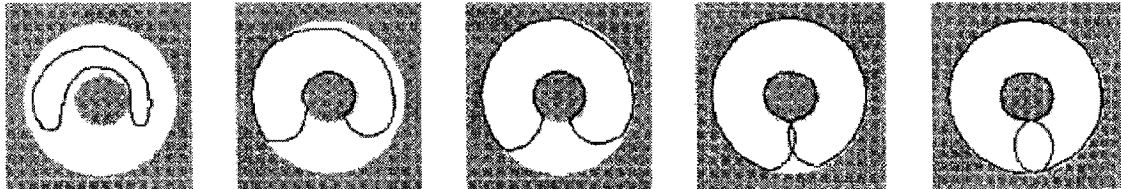
Développé par Kass, Witkin et Terzopoulos (1987), le contour actif (*snake*) est une courbe splinaire qui cherche à atteindre un état d'énergie minimale, attirée par les caractéristiques saillantes d'une image comme les lignes et les contours. C'est un modèle paramétrique explicite, dont la position dans l'espace à deux dimensions est donnée par  $v(s) = (x(s), y(s))$ ,  $s$  étant le paramètre, compris entre 0 et 1. Les déformations du contour actif sont occasionnées par trois types d'énergie de cette façon :

$$E(v(s)) = \int_0^1 E_{in}(v(s)) + E_{im}(v(s)) + E_{con}(v(s)) ds . \quad (1.1)$$

$E_{in}$  est l'énergie interne de déformation contrôlée par un paramètre de tension, qui tend à éliminer les ondulations en réduisant la longueur de la courbe, et par un paramètre de rigidité, qui tend à rendre la courbe plus lisse et moins flexible.  $E_{im}$  est l'énergie associée au contenu de l'image, qui peut attirer la courbe vers les lignes selon l'intensité de l'image et/ou les arêtes selon les gradients de l'image.  $E_{con}$  est l'énergie associée aux contraintes spécifiées par un utilisateur, par exemple des forces spécifiques à certaines régions.

Ce type de modèle déformable est très sensible aux paramètres d'initialisation, telle que sa position initiale. Conçu à l'origine pour des applications interactives, il performe bien dans les applications non interactives s'il est initialisé près de l'objet désiré, mais a tendance à rester piégé près de faux contours. Il éprouve également de la difficulté à épouser les contours d'un objet concave et à modéliser des objets de forme allongée à cause de sa flexibilité géométrique limitée par les contraintes d'énergie interne. Un autre

aspect négatif assez important du contour actif classique est qu'il ne supporte pas les changements de topologie (voir la figure 1.2). Cela empêche la segmentation d'organes présentant des trous, telles que les vertèbres, ou de multiples objets, comme dans le cas des coupes transversales d'embranchements d'une artère.



**Figure 1.2 : Évolution d'un contour actif classique (courbe fermée) pour la segmentation d'un anneau. Les changements topologiques ne sont pas supportés. (Tek et Kimia 1997)**

Une amélioration au contour actif classique a été proposée par Cohen et Cohen (1990) afin de résoudre les problèmes d'instabilité et d'initialisation. Le modèle est une courbe qui se comporte davantage comme un ballon, gonflé par une force interne additionnelle qui agit dans la direction normale. Cette pression interne évite à la courbe de rétrécir sur elle-même et de rester prise dans les zones présentant de fausses arêtes ou du bruit en la poussant à dépasser les contours insignifiants. Toutefois, il faut savoir si le ballon doit se gonfler ou se dégonfler pour atteindre la frontière. Également, l'introduction de la force interne ajoute un paramètre arbitraire additionnel qui influence fortement le résultat de la segmentation.

Xu et Prince (1998) ont plutôt utilisé une nouvelle force externe pour améliorer les problèmes associés à l'initialisation et au manque de convergence pour les frontières concaves. Cette force est la diffusion des vecteurs du gradient des contours en niveaux de gris dérivés de l'image. Contrairement au modèle avec une pression interne, il n'est pas nécessaire de connaître le sens du déplacement du contour actif, car ce dernier peut automatiquement se gonfler ou se dégonfler. En revanche, il a tendance à osciller autour des arêtes et est plus exigeant en termes de temps de calcul. Récemment, Wei, Zhou et

Wan (2004) ont réduit les oscillations et amélioré la rapidité du modèle en utilisant une diffusion non linéaire.

La sélection manuelle des paramètres influençant le comportement du contour actif est difficile et demeure un compromis entre la rigidité du modèle et sa proximité finale envers les limites de l'objet à segmenter.

### 1.2.2.2 Ensemble de niveaux

Une autre approche de modèle déformable consiste à utiliser des formulations implicites qui ont comme avantage de supporter les changements topologiques, contrairement aux formulations explicites (cas du contour actif). De telles formulations ont été introduites à l'aide des ensembles de niveaux (*level sets*) par Caselles, Catté, Coll et Dibos (1993) et Malladi, Sethian et Vemuri (1995), en se basant sur la théorie de l'évolution des contours.

Le modèle est un front qui se déplace temporellement dans un champ de gradient. Sa vitesse  $F$  peut dépendre de plusieurs propriétés :

- locales, reliées par exemple à la courbure et au vecteur normal du front;
- globales, reliées par exemple à la forme et à la position du front;
- externes, indépendantes du front, reliées par exemple au contenu de l'image à segmenter.

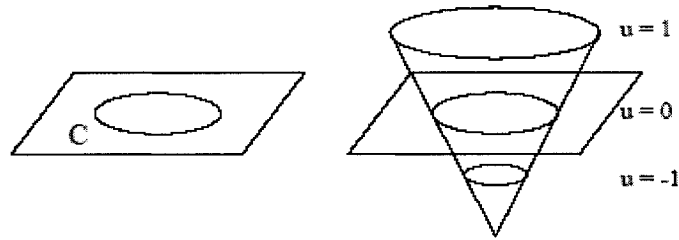
La méthode d'ensemble de niveaux perçoit le déplacement du front dans le temps selon une perspective de valeur initiale. Pour la segmentation en deux dimensions, on considère l'ensemble de niveau zéro, c'est-à-dire la courbe  $C$  (dans  $\mathcal{R}^2$ ) correspondant à la valeur nulle d'une autre fonction  $u$  dans un espace à dimensions plus élevées (par exemple dans  $\mathcal{R}^3$ ), à un temps  $t$  (voir la figure 1.3) :



$$C(t) = \{(x, y) \mid u(x, y, t) = 0\}. \quad (1.2)$$

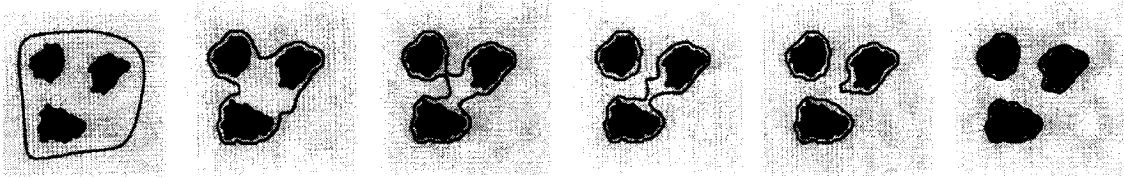
La fonction  $u$  évolue selon la formule suivante :

$$u_t + F|\nabla u| = 0. \quad (1.3)$$



**Figure 1.3 : Ensemble de niveaux : courbe  $C$  (dans  $\mathcal{R}^2$ ) correspondant à la valeur nulle d'une autre fonction  $u$ . (Leventon, Faugeras, Grimson et Wells 2000)**

La figure 1.4 montre un exemple de segmentation d'une image avec cette méthode, qui permet l'adaptation automatique à la topologie des objets : la courbe se divise automatiquement en plusieurs segments pour retrouver la frontière des trois différents objets dans l'image.



**Figure 1.4 : Adaptation topologique de l'ensemble de niveaux. La courbe en noir se divise pour segmenter les divers éléments de l'image. (Malladi, Sethian et Vemuri 1995)**

Pour de plus amples détails sur la méthode d'ensemble de niveaux et de ses applications, le lecteur est invité à consulter le livre de Sethian (1999).

L'absence du concept d'énergie minimisée dans cette formulation peut causer des problèmes liés à l'arrêt de l'évolution de l'ensemble de niveaux. Dans les cas pratiques, un critère d'arrêt est nécessaire, ce qui rend le modèle sensible aux paramètres d'arrêt.

Wang, Taylor, Hsiau, Parker et Dutton (1998) (voir également Wang, Dutton et Taylor 1999) décrivent l'application d'une telle méthode pour la reconstruction de modèles vasculaires 3D à partir d'images produites par résonance magnétique. Une segmentation 2D est réalisée selon des plans perpendiculaires au vaisseau sanguin. Le front est initialement un petit cercle, positionné sur l'axe médial, qui évolue avec le temps. La frontière du système vasculaire est trouvée de cette façon pour chaque image, puis adoucie à l'aide d'un lissage laplacien. Malheureusement, le lissage laplacien ne garantit pas que la forme obtenue soit précise, et les plans perpendiculaires utilisés posent des difficultés aux embranchements des vaisseaux.

### **1.2.2.3 Modèle géodésique**

Le modèle déformable d'ensemble de niveaux peut être inséré dans une approche de minimisation d'énergie en le faisant évoluer selon des calculs géodésiques (courbes de distance minimale), tel que proposé par Caselles, Kimmel et Sapiro (1997). Une métrique, définie par le contenu géométrique de l'image, est utilisée pour faire correspondre l'état d'énergie minimale à la segmentation de l'objet désiré, ce qui évite le besoin de conditions d'arrêt spéciales. Le modèle peut se diviser et fusionner, ce qui permet la segmentation de multiples objets. Moins de paramètres sont nécessaires par rapport à la formulation des ensembles de niveaux classiques, mais cela a pour conséquence que le modèle géodésique est plus fortement dépendant de la position initiale du modèle.

Niessen, Romeny et Viergever (1998) ont proposée une extension au modèle déformable géodésique. Cette méthode, à la différence du modèle géodésique précédent, considère plusieurs ensembles de niveaux dans l'image, en conservant l'aspect de minimisation de l'énergie. C'est une autre approche permettant de segmenter plusieurs objets simultanément. Elle requiert moins de paramètres que les ensembles de niveaux classiques. Toutefois, elle ne peut s'appliquer que pour des images dans lesquelles

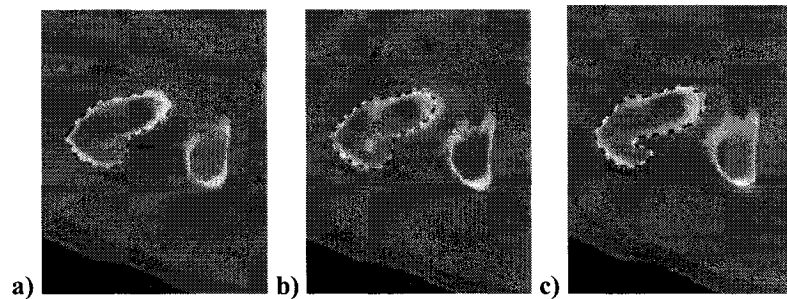
l'intensité est grossièrement constante pour chaque objet, de même que pour des objets dont les frontières sont fermées ou peuvent être fermées par diffusion non linéaire.

#### **1.2.2.4 Modèle probabiliste/statistique**

Une façon d'incorporer des connaissances a priori dans le modèle déformable est d'utiliser des distributions de probabilité basées sur des données statistiques pour améliorer le processus de correspondance du modèle avec l'image. Dans cet ordre d'idée, Paragios et Deriche (1999), dans le cadre du repérage et suivi d'objets non rigides à travers une séquence d'images, ont proposé l'utilisation d'un modèle géodésique basé sur une modélisation statistique supervisée comprenant les connaissances a priori sur les frontières désirées.

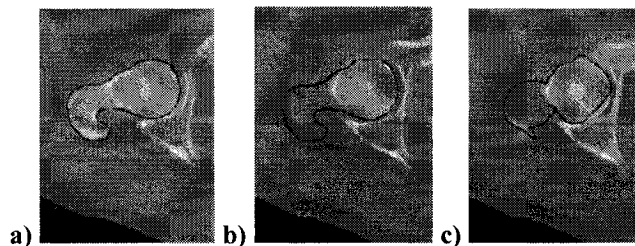
Une version améliorée (généralisée) du contour actif classique pour effectuer une segmentation de séquences d'images spatiales et temporelles sous forme de contour actif discriminant statistique a été développée par Pardo, Radeva et Cabello (2003). Ce modèle, qui peut être considéré comme une méthode  $2\frac{1}{2}D$ , effectue un apprentissage supervisé des frontières de l'objet dans une image de la série pour améliorer la segmentation de la prochaine image. Chaque partie de la frontière de l'objet est projetée dans l'espace des caractéristiques (qui représente toutes les caractéristiques d'intérêt des images), généré par une banque de filtres gaussiens. Un classificateur paramétrique est généré à partir des contours des images et joint dans le processus de minimisation d'énergie du modèle pour guider la déformation de ce dernier dans l'image suivante. Le modèle sélectionne et classifie les caractéristiques de l'image à l'aide du classificateur paramétrique et se déforme pour minimiser la différence entre les informations apprises et l'image. Le choix d'un classificateur continu le long des contours permet de filtrer le bruit sans produire un lissage excessif. Comparativement au contour actif traditionnel, le modèle développé permet de mieux capter les formes d'objets concaves (voir la figure

1.5). Il n'a pas de but prédéfini et apprend les caractéristiques des contours désirés pour chaque problème de segmentation.



**Figure 1.5 : Segmentation avec un contour actif discriminant statistique. En a), la frontière trouvée en ligne pointillée. Cette frontière sert de frontière initiale en b) pour la segmentation de l'image suivante. Résultat final de la segmentation en c). La portion concave de l'objet est bien captée. (Pardo, Radeva et Cabello 2003)**

Un problème se présente lorsque les caractéristiques dans la géométrie et la texture changent dramatiquement d'une image à l'autre, car la segmentation échoue, tel qu'illustré sur la figure 1.6. Dans ces cas, l'interaction d'un usager est nécessaire.



**Figure 1.6 : Problème de segmentation avec le contour actif discriminant statistique. La frontière référence en a) est utilisée comme frontière initiale en b), ce qui donne le mauvais résultat présenté en c). (Pardo, Radeva et Cabello 2003)**

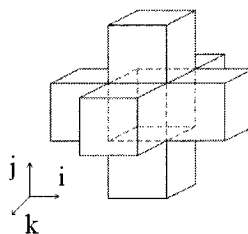
### 1.2.3 Croissance de région

La croissance de région (*region growing*) constitue un autre type de technique de segmentation, qui se base sur la connectivité topologique des éléments des images. Elle est la duale de la segmentation par détection de contours : elle recherche les régions aux caractéristiques communes et non pas uniquement la limite de celles-ci. Elle peut être appliquée en trois dimensions mais, contrairement aux modèles déformables

tridimensionnels, elle doit être suivie d'une étape de reconstruction. C'est pourquoi elle appartient à la première grande approche de modélisation géométrique d'images médicales, celle en deux étapes.

Considérant la série d'images médicales (coupes transversales) comme un volume, les pixels deviennent des éléments de volume (voxels). Selon cette méthode, les voxels sont classifiés selon s'ils se trouvent à l'intérieur ou à l'extérieur de l'objet, de façon à produire une région connectée représentant l'organe désiré. La segmentation part d'un germe (un voxel source sur l'une des images) qui fait partie de la région. D'autres voxels voisins sont progressivement associés à la région selon un ou plusieurs critères. L'intensité, le contraste et l'entropie constituent des exemples pouvant être utilisés comme critères de sélection (voir Maître 2003). En plus de ces critères, la classification se base sur la position des voxels en relation avec leurs voisins.

Cebral et Löhner (2001) ont proposé une méthode de croissance de région similaire à une avance de front dans le cadre de la segmentation d'images médicales pour des modélisations d'écoulement sanguin dans les vaisseaux. À partir d'un voxel source spécifié par l'utilisateur, l'algorithme marque tous les voxels connectés au voxel source ayant une intensité située dans la même plage. Cette connectivité se fait par l'entremise des six voisins immédiats d'un voxel, c'est-à-dire les quatre voisins dans le plan image et les deux voxels correspondant sur l'image précédente et l'image suivante, tel qu'illustré à la figure 1.7.



**Figure 1.7 : Voisins immédiats d'un voxel dans l'algorithme de croissance de région. Les voisins d'un voxel (en pointillés rouges) situé à  $(i,j,k)$  sont les voxels situés à  $(i-1,j,k)$ ,  $(i+1,j,k)$ ,  $(i,j-1,k)$ ,  $(i,j+1,k)$ ,  $(i,j,k-1)$  et  $(i,j,k+1)$ . Les images sont dans le plan  $i-j$  et la dimension  $k$  représente différentes images.**

Cette approche nécessite l'intervention humaine et a pour difficulté le choix d'un bon voxel de départ et d'une plage d'intensités qui permettra de récupérer tous les voxels faisant partie de l'organe, mais pas davantage. De plus, à cause du bruit sur les données de départ, la connectivité d'une région représentant un même vaisseau sanguin peut être incomplète. La croissance de région a comme avantage, tout comme les modèles d'ensembles de niveaux, de supporter les changements topologiques. Elle peut être améliorée en la combinant avec l'utilisation de prétraitements sur les images et de post-traitements (par exemple des filtres statistiques pour remplir les trous indésirables), tel que proposé par Cebal et Löhner (2001).

### **1.3 Reconstruction 3D à partir de segments 2D**

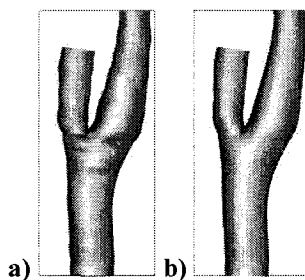
Une fois les images médicales segmentées une à une, l'étape suivante est de reconstituer un modèle tridimensionnel à partir des données en deux dimensions provenant de la segmentation, généralement les contours de l'objet. Il s'agit donc de connecter la séquence de contours 2D en une surface 3D continue. Ce modèle tridimensionnel peut être paramétrique, représenté par exemple par des splines, ou encore polygonal, formé d'un agencement de polygones. Ce dernier semble davantage utilisé dans les applications médicales.

Faire la correspondance entre deux points situés sur deux contours successifs n'est pas trivial. Différentes méthodes ont été utilisées pour faire la reconstruction de modèles 3D à partir des objets segmentés en 2D. Les sections suivantes discutent de quelques-unes de ces méthodes : l'approximation / interpolation par des splines dans la catégorie des surfaces paramétriques; les cubes baladeurs, la fragmentation directe et le flux optique dans la catégorie des surfaces polygonales.

### 1.3.1 Approximation / interpolation par des splines

Dans la catégorie des modèles de reconstruction de surfaces paramétriques, un modèle simple et naturel est d'utiliser une approximation basée sur le plus proche échantillon ou sur une moyenne des échantillons voisins. Cette approximation locale peut être constituée, par exemple, de surfaces polynomiales par morceaux, telle que des facettes splinaires. Les échantillons utilisés pour l'approximation proviennent des segments 2D trouvés pour chaque image lors de la phase de segmentation. L'approximation par des splines crée une surface optimale qui ne passe pas nécessairement par les échantillons, c'est-à-dire par les contours de l'organe. Le modèle est relativement lisse mais peut être inadéquat dans les régions de haute courbure.

La reconstruction de la surface peut également se faire par interpolation par des splines. Dans ce cas, la surface passe obligatoirement par les échantillons mais son aspect lisse n'est pas garanti (voir la figure 1.8a). Moore, Steinman, Holdsworth et Ethier (1999) ont comparé différents modèles de reconstruction d'artères par approximation et interpolation. Ils ont déterminé que l'approximation est la méthode qui performe le mieux si elle est précédée d'une « augmentation » des données originales par l'introduction de contours 2D supplémentaires entre les images trouvés par interpolation linéaire (voir la figure 1.8b). Ces nouveaux contours n'ajoutent pas d'information additionnelle mais forcent la surface reconstruite par approximation à correspondre davantage aux données originales.



**Figure 1.8 : Reconstruction de la surface paramétrique d'une artère avec des splines par interpolation en a) et par approximation précédée d'une augmentation des données en b). (Moore, Steinman, Holdsworth et Ethier 1999)**

Le domaine de la reconstruction à partir de données tridimensionnelles à l'aide de modèles paramétriques est en ébullition et de nouvelles approches utilisant des variantes de splines font régulièrement leur apparition. À titre d'exemple, Rössl, Zeilfelder, Nürnberger et Seidel (2004) ont développé une approche de reconstruction de modèles continus à partir d'échantillons tridimensionnels en utilisant des supers splines quadratiques trivariées sur une partition tétraédrique uniforme.

Pour l'application de simulations d'écoulement, le modèle tridimensionnel doit être transformé en maillage. La reconstruction avec une surface paramétrique n'est alors pas indiquée car elle nécessiterait une étape de conversion vers un modèle polygonal. Elle est davantage adaptée à la visualisation de modèles.

### 1.3.2 Cubes baladeurs

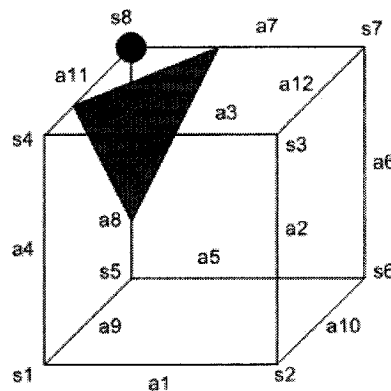
La reconstruction de surface peut se baser sur des isosurfaces, c'est-à-dire des surfaces composées de points ayant la même valeur numérique (dans le cas présent, la même valeur d'intensité). Suivant cette idée, Lorensen et Cline (1987) ont développé un algorithme appelé « cubes baladeurs » (*marching cubes*) qui crée un modèle polygonal composé de triangles avec des surfaces à densité constante à partir de données médicales 3D. La localisation de la surface est effectuée à partir d'une valeur spécifiée par l'utilisateur. La segmentation n'est pas nécessaire comme étape précédente si l'organe à modéliser est le seul d'intensité égale à l'isovaleur, mais elle peut automatiser le choix de cette isovaleur, qui peut simplement correspondre à la valeur moyenne des contours trouvés par la segmentation.

L'idée est de localiser la surface en utilisant un cube logique dont les faces du dessous et du dessus correspondent à deux images successives et les sommets à huit pixels, quatre de chaque image. L'algorithme détermine la façon dont la surface est intersectée par le cube, puis « marche » jusqu'au prochain cube. Il fonctionne de la façon suivante :



1. Créer un cube à partir de deux images adjacentes, avec 4 sommets appartenant à la première image et 4 à la seconde.
2. Attribuer une valeur pour chaque sommet selon s'il se trouve à l'extérieur de la surface (0), ou à l'intérieur ou sur la surface (1), en comparant la valeur de densité des sommets avec celle de la surface.
3. Calculer un index pour chaque cube selon les valeurs des sommets, qui correspond à l'un des 256 états possibles (2 possibilités de valeur pour chacun des 8 sommets).
4. Trouver les arêtes du cube intersectées par la surface (configuration du cube) à partir de l'index, à l'aide d'une table précalculée contenant une liste d'arêtes formant les triangles de la surface.
5. Interpoler linéairement la position d'intersection de la surface le long des arêtes à partir des valeurs des deux sommets délimitants.

La figure 1.9 montre un exemple du résultat de l'intersection de l'isosurface avec un cube baladeur.



**Figure 1.9 : Algorithme des cubes baladeurs : exemple de configuration possible de l'intersection de la surface de l'objet avec le cube logique. Le sommet s8 étant le seul à faire partie de l'intérieur de l'objet, le triangle rouge est donc créé, chacun de ses sommets étant positionnés par interpolation linéaire sur les arêtes entre s8 et ses trois sommets voisins. L'index dans ce cas est 10000000.**

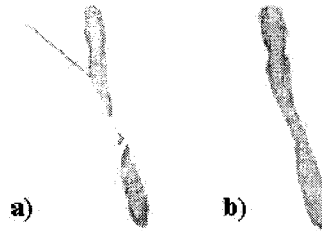
Des cas ambigus peuvent survenir dans l'éventualité de plusieurs intersections au sein du même cube logique, mais des configurations additionnelles ont été ajoutées pour

limiter ce problème. L'algorithme des cubes baladeurs demeure une référence dans le domaine de la visualisation et de la reconstruction de surfaces.

Matsushita et Kaneko (2000) ont dénoté que l'algorithme des cubes baladeurs ne préserve pas le volume des données originales. Cette distorsion est remarquable pour les petits objets (voir la figure 1.10a). Afin de pallier à ce problème, ils ont proposé une amélioration pour convertir des données 3D de volume telles que des images médicales en surfaces qui préservent localement le volume original. Leur algorithme est présenté pour des objets 2D et 3D et fonctionne comme ceci :

1. Reconstruire les surfaces en utilisant l'algorithme des cubes baladeurs avec les valeurs originales des images.
2. Pour chaque voxel, calculer le ratio du volume appartenant à l'intérieur et celui appartenant à l'extérieur de l'organe à reconstruire, puis calculer la nouvelle valeur du voxel pondérée selon ces proportions.
3. Calculer la différence entre la valeur originale et la nouvelle valeur pondérée des voxels. Si cette différence est positive et supérieure à un seuil, incrémenter la valeur de travail du voxel (initialisée à la valeur originale). Si cette différence est négative et inférieure à un seuil, décrémenter la valeur de travail du voxel.
4. Reconstruire les surfaces en utilisant les cubes baladeurs avec les valeurs de travail des voxels.
5. Répéter les étapes 2 à 4 jusqu'à ce que la différence trouvée en 3) soit acceptable.

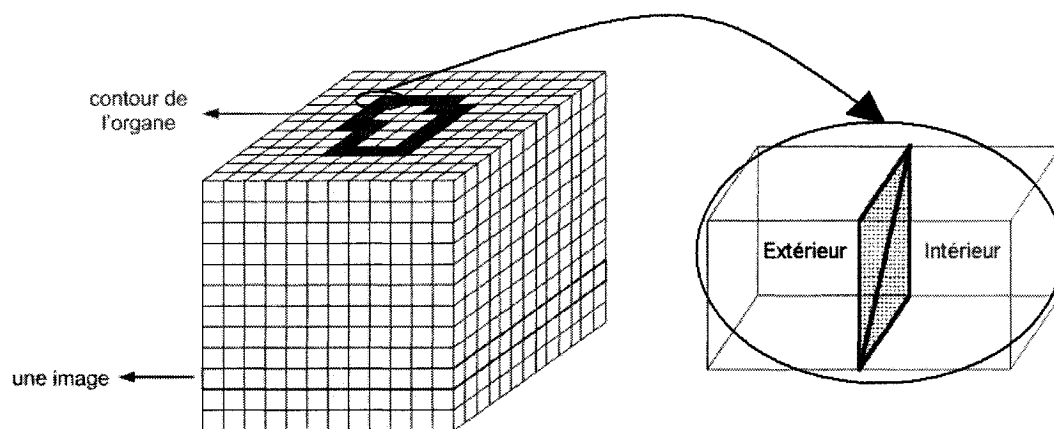
Généralement, 10 à 20 itérations sont suffisantes. Cet algorithme est très exigeant en termes de temps de calcul. Pour alléger le calcul, les trois modèles les plus fréquents des voxels frontaliers sont utilisés pour calculer les ratios internes / externes. Selon les auteurs, l'algorithme est supérieur aux cubes baladeurs particulièrement pour les petits objets, près de l'échelle de résolution (voir la figure 1.10b).



**Figure 1.10 : Reconstruction d'un vaisseau sanguin a) par l'algorithme des cubes baladeurs, b) par l'algorithme de préservation du volume. En a), le volume du vaisseau n'est pas préservé. (Matsushita et Kaneko 2000)**

### 1.3.3 Fragmentation directe

La reconstruction de la surface polygonale peut se baser alternativement sur la forme géométrique qu'on peut attribuer à un voxel, c'est-à-dire un cube. À partir des images médicales segmentées, un volume est constitué en les empilant. Les pixels deviennent ainsi des voxels, qu'on peut assimiler à des cubes de dimensions réelles. Dans les travaux de Cebal et Löhner (2001), la surface est créée en générant un ensemble de triangles situés sur les faces externes des voxels constituant le contour de l'objet (voir la figure 1.11), dans un processus appelé « fragmentation directe des voxels frontaliers » (*direct tessellation*).



**Figure 1.11 : Fragmentation directe des voxels frontaliers. Deux (ou quatre) triangles sont créés sur les faces externes des voxels constituant le contour de l'objet, c'est-à-dire à la limite entre l'extérieur et l'intérieur de l'objet.**

Ce processus produit une surface « en escaliers » à l'échelle du voxel car tous les triangles sont parallèles aux faces des voxels, elles-mêmes parallèles aux axes du système de référence, tel qu'illustré à la figure 1.12a. Il est donc nécessaire de lisser la surface obtenue, peu importe son utilisation (voir la figure 1.12b). Les auteurs ont arrêté leur choix sur une méthode de lissage présentée par Taubin (1995a, 1995b). Le lissage de surfaces est introduit à la section 1.5.

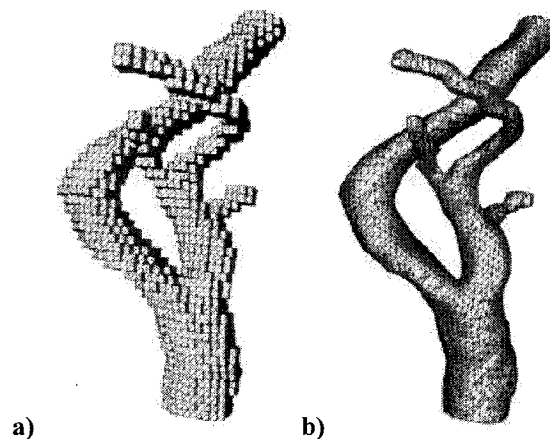


Figure 1.12 : Reconstruction de la carotide avec la fragmentation directe. La surface en escaliers résultante en a) est lissée en b). (Cebal et Löhner 2001)

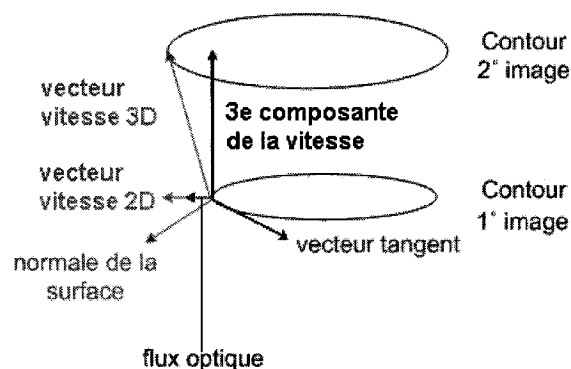
### 1.3.4 Flux optique

Le déplacement du scanneur par rapport à l'objet scanné lors de l'acquisition des images médicales sous forme de coupes peut être utilisé pour introduire le concept de mouvement dans le processus de reconstruction d'un modèle 3D polygonal de l'objet à partir de segments 2D. Même si l'objet à modéliser ne bouge pas, il y a un mouvement apparent entre les différentes images, visible par le changement d'intensité d'une image à l'autre pour un même pixel<sup>1</sup>.

<sup>1</sup> Dans les cas d'acquisition d'images médicales *in vivo*, les organes peuvent subir une déformation dans le temps, due entre autres à la respiration. Il n'est pas question de ce type de mouvement dans la méthode du flux optique.

Weng, Yang et Pierson (1997) ont proposé une méthode pour reconstruire un modèle 3D à partir d'images médicales en tirant profit de ce mouvement apparent, analysé à l'aide du flux optique. Il est possible de retrouver le mouvement 3D d'un objet à partir du champ de flux optique en ajoutant des contraintes additionnelles, et d'extraire de l'information sur la surface de l'objet à partir du mouvement 3D. La technique comprend les étapes suivantes (voir la figure 1.13) :

1. Calcul du champ de flux optique entre les contours de l'organe de deux images segmentées consécutives par une approche basée sur les dérivées de l'intensité des images (gradient) et la méthode des moindres carrés.
2. Calcul des vecteurs de vitesse en 2D (plan image) à partir du flux optique et du mouvement apparent des pixels entre paires d'images.
3. Calcul des vecteurs de vitesse en 3D en assumant que la troisième composante est donnée par le déplacement du scanneur (connu).
4. Calcul des vecteurs tangents des segments 2D.
5. Calcul des vecteurs normaux de la surface par le produit vectoriel des vecteurs tangents 2D et des vecteurs de vitesse 3D.
6. Reconstruction de l'objet avec une forme par défaut dictée par les normales.



**Figure 1.13 : Éléments utilisés dans l'algorithme de reconstruction de surfaces polygonales basé sur le flux optique.**

La méthode est toutefois limitée car beaucoup de calculs sont nécessaires pour le flux optique, l'exactitude du flux optique 2D affecte énormément la reconstruction de la

forme 3D et l'algorithme assume que le champ de vitesse pour chaque image a été calculé de façon précise.

## **1.4 Segmentation 3D**

Les sections précédentes ont présenté des méthodes qui s'inscrivent dans la première approche de modélisation 3D d'un organe à partir d'images médicales : la segmentation 2D des données suivie d'une reconstruction 3D de la surface de l'objet à partir des segments 2D. Cette section fait état des idées et travaux développés dans l'optique de la seconde approche, c'est-à-dire la segmentation 3D des données, qui permet l'extraction directe d'un modèle 3D sans passer par une étape de reconstruction.

La plupart des techniques de segmentation 3D sont des extensions des versions 2D, tels que les modèles déformables. Certaines tirent profit des connaissances d'autres disciplines de la science, comme la dynamique des particules ou la propagation d'ondes. Les sections suivantes présentent un survol de ces méthodes de segmentation 3D : la détection de contours, les modèles déformables (surface active et ballon, ensemble de niveaux, modèle de forme adaptée à l'organe, bulles 3D, mise en correspondance de surfaces), les particules dynamiques et la propagation d'ondes.

### **1.4.1 Détection de contours**

De même qu'en deux dimensions, il existe des opérateurs tridimensionnels de bas niveau qui détectent les contours des objets en se basant sur les différences d'intensité ou gradient. La version 3D de l'opérateur Canny (présenté à la section 1.2.1) en est un exemple. Ces opérateurs présentent les mêmes désavantages que dans leur version 2D : ils ne peuvent gérer la présence de bruit dans les données et ne tirent pas profit des propriétés globales ou connues des objets à segmenter. La détection des contours d'un

objet uniquement par ce type d'opérateur 3D dans le domaine médical n'est pas adéquate mais peut tout de même servir comme approximation dans des étapes intermédiaires.

### **1.4.2 Modèles déformables**

Introduits dans une version bidimensionnelle, les modèles déformables peuvent être étendus à la troisième dimension. Ils possèdent les mêmes caractéristiques et comportements que leurs homologues 2D, c'est-à-dire qu'ils se déforment sous l'action de forces pour ultimement délimiter les frontières des objets 3D.

Dans les prochaines sections, le lecteur trouvera une présentation de quelques-uns des modèles déformables tridimensionnels utilisés pour modéliser des organes à partir d'images médicales : surface active et ballon, ensemble de niveaux, modèle de forme adaptée à l'organe, bulles 3D et mise en correspondance de surfaces.

#### **1.4.2.1 Surface active**

Le contour actif classique bidimensionnel, ou sa version amélioré par Cohen et Cohen (1990) qui se comporte comme un ballon, cherchant à épouser les contours d'objets de façon à minimiser son énergie, peut être étendu à la segmentation tridimensionnelle. Il ne s'agit plus d'une courbe évoluant sur chaque image séparément mais d'une surface évoluant sur l'ensemble des images en 3D.

Plusieurs chercheurs ont repris l'idée du ballon 2D et l'ont utilisée pour la segmentation 3D d'images médicales. Cohen et Cohen (1993) ont raffiné leur modèle original en développant les équations le régissant pour faire une généralisation en 3D. À cause de la complexité des calculs engendrés par le passage au 3D, ils ont utilisé une méthode

d'éléments finis pour résoudre le problème de minimisation d'énergie de la surface, qui est plus stable et qui converge plus rapidement que d'autres méthodes telles que les différences finies. Ils ont appliqué leur approche à la segmentation d'images produites par résonance magnétique.

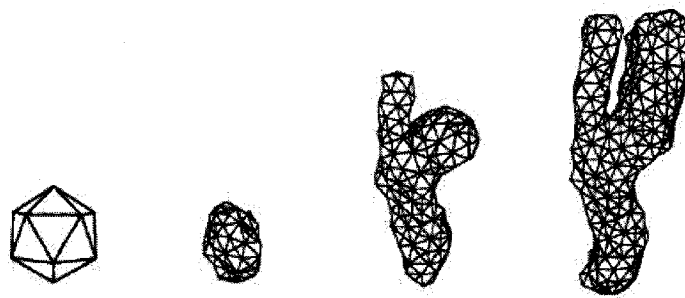
Chen et Medioni (1995) ont utilisé le modèle du ballon dans le but de construire la surface d'un objet à partir d'un ensemble d'images télémétriques. Dans une image conventionnelle, chaque pixel possède une valeur d'intensité qui est reliée indirectement à la géométrie de la surface des objets de la scène représentée. Dans une image télémétrique, chaque pixel exprime la distance entre un cadre de référence connu et un point visible de la scène, reproduisant ainsi la structure 3D de la scène. Les données de départ servant à l'obtention d'un modèle 3D étant différentes, l'utilisation de telles images modifie légèrement la façon de vérifier si le modèle a atteint les contours de l'objet. Les auteurs ont basé leur approche sur un modèle de ballon dynamique représenté par un maillage de triangles, dont les sommets sont reliés à leur voisin à travers des ressorts afin de simuler la tension de surface et garder la surface lisse. Pour que les triangles demeurent également distribués au fur et à mesure que le ballon se gonfle, le système inclut un schéma de subdivision locale. Puisque l'approche n'est pas basée sur la minimisation globale, la surface peut s'adapter aux caractéristiques locales de l'objet à segmenter. Toutefois, comme les autres modèles de ballon, les changements de topologie ne sont pas supportés.

En s'inspirant, entre autres, des travaux de Chen et Medioni (1995), Ladak, Milner et Steinman (2000) ont proposé une technique semi-automatique de segmentation 3D rapide appliquée à la bifurcation des carotides, à partir d'images de résonance magnétique, utilisant un ballon virtuel gonflant. Les étapes concernant la segmentation 3D sont les suivantes :



1. Interpolation : les images sont assemblées en un bloc 3D et 5 images supplémentaires sont linéairement interpolées dans la direction axiale entre chaque paire d'images sources adjacentes.
2. Segmentation 3D :
  - a. Placement du ballon initial à l'intérieur de l'artère par un utilisateur.
  - b. Inflation automatique du ballon comprenant une force basée sur l'image, c'est-à-dire basée sur l'intensité des voxels situés en dessous d'un certain seuil, et une tension de surface simulée pour conserver l'aspect lisse. Le maillage du ballon est raffiné progressivement selon la longueur des arêtes (voir la figure 1.14).
  - c. Localisation automatique de la paroi artérielle : déformation additionnelle du ballon par un champ de force basé sur la localisation des arêtes.

L'objectif de la méthode proposée est de réduire le temps et les efforts requis pour produire des maillages d'éléments finis, utiles pour les simulations d'écoulement sanguin. La méthode se veut plus rapide que la segmentation 2D traditionnelle suivie de la reconstruction 3D, car elle requiert l'intervention humaine pour situer le modèle de départ une seule fois et non pas pour chacune des images. Toutefois, il n'est pas évident que la géométrie reconstruite représente mieux l'anatomie réelle que dans l'approche 2D et les changements de topologie ne peuvent être traités.



**Figure 1.14 : Évolution d'un ballon 3D dans le processus de segmentation de la bifurcation des carotides. (Gill, Ladak, Steinman et Fenster 2000)**

Les travaux de Gill, Ladak, Steinman et Fenster (2000) sur une technique semi-automatique de segmentation des carotides à partir d'images médicales produites par

ultrasons ressemblent à ceux de Ladak, Milner et Steinman (2000), mais comportent en plus une analyse détaillée de la précision et de la variabilité de la méthode. Puisque l'initialisation du ballon se fait manuellement, il y a une variabilité de la position et de la forme de la frontière finale trouvée. Les auteurs ont déterminé que leur approche semi-automatique a une moins grande variabilité, qui dépend de la position initiale du ballon, que la variabilité intra-spécialiste pour une segmentation manuelle, tout en produisant généralement un modèle final en accord avec celui résultant de la segmentation manuelle.

Un problème d'auto-intersection du ballon gonflant se présente parfois dans les régions de haute courbure, dépendamment de la taille des triangles du modèle, des forces régissant le ballon et du choix des paramètres du modèle. Dans ces cas, le ballon se met à gonfler intérieurement, ce qui n'est pas le comportement souhaité.

Jiang et Fan (2002) ont repris l'idée du ballon 3D basé sur les méthodes d'éléments finis et l'ont intégrée dans un algorithme génétique. Leurs travaux utilisent un algorithme de segmentation 3D et de recalage pour les images médicales, qui tente de réduire le temps de calcul par rapport aux méthodes d'optimisation de problèmes en utilisant la « parallélisation ». L'approche est en deux étapes :

1. Les surfaces des objets sont trouvées par une méthode aux différences finies basée sur des équations dynamiques. Cela donne une description grossière mais rapide.
2. Les surfaces sont ensuite raffinées en les passant comme première génération à un processus de minimisation d'énergie implantée par un algorithme génétique parallèle.

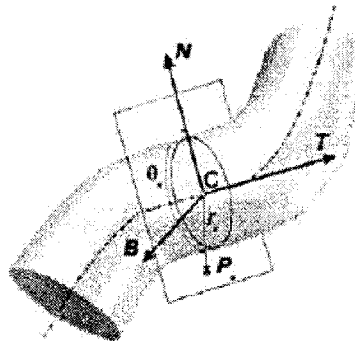
Cette méthode de segmentation 3D a été testée par les auteurs pour les ventricules latéraux du cerveau et semble encourageante mais souffre des mêmes inconvénients que

les autres modèles de ballon, notamment au niveau de l'absence d'adaptation à la topologie.

#### 1.4.2.2 Modèles cylindriques et tubulaires

Des modèles déformables tridimensionnels ont été développés spécifiquement pour la modélisation de vaisseaux sanguins afin de pallier aux problèmes des modèles plus généraux, particulièrement pour les régions de bifurcations. Ces développements ont principalement été réalisés dans le cadre de l'évaluation de maladies vasculaires, par exemple pour la mesure du degré de sténose des veines à partir d'images produites par angiographie par résonance magnétique (ARM).

Frangi, Niessen, Hooijgeveen, van Walsum et Viergever (1999) ont élaboré un modèle déformable basé sur une surface paramétrée de façon cylindrique. Ce modèle consiste en une représentation de l'axe central du vaisseau, couplée à la surface de la paroi du vaisseau (voir la figure 1.15).



**Figure 1.15 : Modèle déformable cylindrique, représenté par un axe central et une paroi. (Frangi, Niessen, Hooijgeveen, van Walsum et Viergever 1999)**

L'axe et la surface sont représentés par une courbe et des surfaces actives, respectivement, et sont déterminés par un processus de minimisation d'énergie (voir la

section précédente sur la surface active). La segmentation tridimensionnelle se fait en deux étapes :

1. Trouver la représentation de l'axe central tout en estimant le diamètre du vaisseau à partir des données fournies par l'angiographie.
2. Modéliser la paroi du vaisseau en utilisant l'axe trouvé précédemment comme paramètre longitudinal.

Le modèle cylindrique possède une forme appropriée pour modéliser des vaisseaux sanguins, et cela a pour conséquence de produire une surface plus lisse que les modèles déformables généraux. Toutefois, à cause de l'emploi de fonctions paramétriques linéaires, les contraintes utilisées pour que les points de la surface soient également espacés tendent à surcomprimer les vaisseaux.

Dans le but de résoudre ce problème, Yim, Cebal, Mullick, Marcos et Choyke (2001) ont développé un modèle déformable qui emploie plutôt un système référentiel tubulaire avec des fonctions paramétriques radiales. Ce système diffère du système cylindrique par l'accommodation d'axes courbés. Il supporte les variations dans le diamètre des vaisseaux et dans la forme des vaisseaux sur les coupes transversales et permet de représenter les détails à plus petite échelle. Par contre, l'intervention humaine est nécessaire dans le processus de définition de l'axe du vaisseau, ce qui rend les modèles cylindriques et tubulaires moins intéressants pour la conception d'une technique de modélisation automatisée. Également, ils ne peuvent supporter les changements de topologie, mais pour les tests effectués par les auteurs, cette considération ne s'applique pas. Enfin, dans les régions de bifurcations, plusieurs modèles sont nécessaires pour représenter la division en plusieurs branches (tubes).

#### **1.4.2.3 Ensemble de niveaux**

Le modèle d'ensemble de niveaux, présenté en 2D à la section 1.2.2.2, peut être appliqué pour la segmentation tridimensionnelle d'images médicales. Le front qui se déplace n'est

plus une courbe dans  $\mathbb{R}^2$  mais une surface dans  $\mathbb{R}^3$ , correspondant à la valeur nulle d'une fonction dans un espace à dimensions plus élevées, par exemple dans  $\mathbb{R}^4$ . L'avantage d'un tel modèle réside dans le support des changements topologiques.

Malladi, Kimmel, Adalsteinsson, Sapiro, Caselles et Sethian (1996) ont travaillé sur les modèles déformables implicites et présentent une approche pour détecter, représenter et mesurer des données médicales 3D. La technique se base sur des surfaces 3D, représentées par des ensembles de niveaux, qui se déforment vers les objets médicaux, selon des mesures géométriques intrinsèques aux données. La représentation de l'objet médical 3D, une surface minimale pondérée dans un espace riemannien, s'inspire des contours actifs géodésiques. La surface minimale est calculée en utilisant la méthode d'ensemble de niveaux à partir de mesures géométriques de l'image, utilisant seulement les points très près de la région d'intérêt (technique à bande étroite ou *narrow-band*) afin de réduire les temps de calculs. Selon les auteurs, la méthode est assez précise pour permettre de calculer l'aire de la surface et le volume des objets détectés.

L'approche à bande étroite (voir Sethian 1999) a été développée dans le but de limiter le nombre de calculs nécessaires pour la mise à jour du déplacement du front. Puisque les calculs se font dans un espace de dimensions plus élevées que la courbe ou la surface se déplaçant, la complexité de l'algorithme est très importante. Si l'on considère l'ensemble de niveau zéro, il est possible de n'utiliser que les points se trouvant dans son voisinage immédiat pour déterminer sa nouvelle position, tel qu'illustré par la figure 1.16.

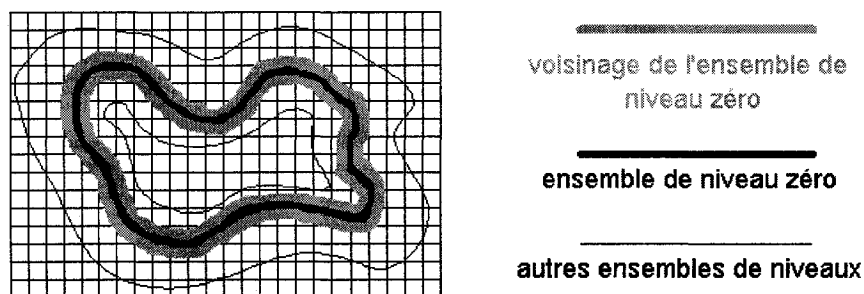
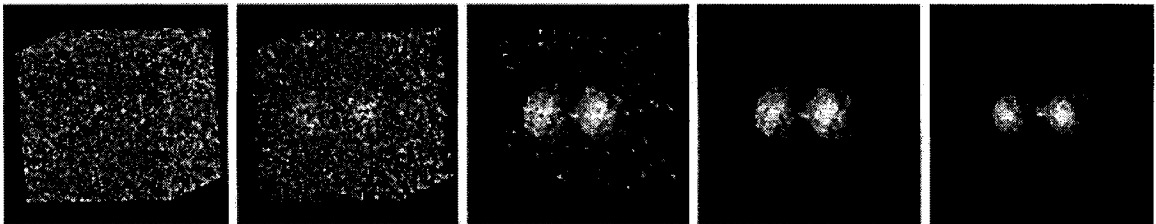


Figure 1.16 : Méthode à bande étroite pour les ensembles de niveaux.

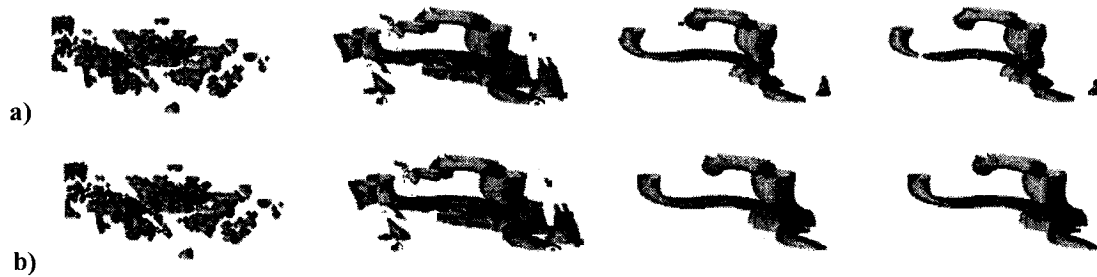
#### 1.4.2.4 Bulles 3D

Une autre stratégie de segmentation 3D d'un objet à partir d'images médicales est d'utiliser non pas un seul modèle déformable mais une panoplie, pouvant se fusionner et se diviser pour mieux épouser la géométrie de l'objet. Tek et Kimia (1997) ont présenté une segmentation 3D en utilisant des descriptions d'objets très simples, des bulles, qui sont initialisées et placées aléatoirement dans les régions homogènes du volume d'images médicales. Elles évoluent selon les informations des images pour retrouver une description complète des formes contenues, tel qu'illustré à la figure 1.17. Elles grossissent et fusionnent pour former de plus grosses bulles ou se divisent pour former de plus petites bulles. Elles se déforment pour adhérer aux frontières des objets sous l'influence des gradients de l'image.



**Figure 1.17 : Évolution des bulles 3D dans le processus de segmentation d'une haltère. (Tek et Kimia 1997)**

Les bulles évoluent dans un espace de « réaction-diffusion ». Le processus de réaction capture les structures à un niveau détaillé à partir des informations complètes car elle favorise la séparation des bulles, alors que le processus de diffusion sert à régulariser et à capturer les structures à un niveau grossier à partir des informations incomplètes, présentant par exemple des trous (voir la figure 1.18), car elle défavorise la séparation des bulles.



**Figure 1.18 : Segmentation 3D avec des bulles. La séquence d'évolution correspondant aux processus de réaction (a) et de diffusion (b). (Tek et Kimia 1997)**

Cette technique offre un plus grand degré d'automatisation que les modèles déformables nécessitant l'initialisation du modèle par un usager, sans toutefois empêcher la possibilité d'interaction. Par contre, puisque les bulles sont représentées à l'aide d'ensembles de niveaux, leur implantation est coûteuse en termes de calculs.

#### **1.4.2.5 Mise en correspondance de surfaces**

La segmentation tridimensionnelle d'images médicales peut se baser sur la mise en correspondance de surfaces. Dans une telle méthode, on tente de faire correspondre un modèle géométrique ressemblant à l'objet à segmenter avec ce dernier. Staib et Duncan (1996) ont intégré cette idée dans une approche dite de recherche de frontière.

Ces auteurs décrivent une méthode permettant de trouver les frontières d'objets dans des images médicales 3D, basée sur des modèles de formes paramétriquement déformables, tirant également profit de l'information préalablement connue sur la forme. La paramétrisation décompose la surface en fonctions de base sinusoïdales. Quatre types de surfaces sont modélisés :

1. tores (tubes fermés);
2. surfaces ouvertes (avec une arête);
3. tubes (surfaces ouvertes avec deux arêtes);
4. surfaces fermées (aucune arête).

La recherche de la frontière est formulée comme un problème d'optimisation utilisant des modèles paramétriques de surfaces de Fourier. Le modèle est apparié à l'image en optimisant, dans l'espace paramétrique, sa correspondance avec une mesure de frontière appliquée à l'image. Des distributions de probabilité sur les paramètres de la représentation sont incorporées pour influencer les modèles vers une tendance de forme globale particulière, tout en permettant les déformations.

Le choix de fonctions de base sinusoïdales a comme limitation de ne pas pouvoir représenter très bien des transitions abruptes ou des frontières complexes parce que cela requiert un nombre important de paramètres. Il y a également des possibilités que l'étape d'optimisation locale fasse converger vers un maximum erroné. Selon les auteurs, les expérimentations ont montré que la méthode est relativement insensible au bruit, aux problèmes de frontières avec des trous et de fausses frontières d'objets voisins, et la dépendance entre l'erreur et la position initiale du modèle est faible. Par contre, pour obtenir l'information préalablement connue, il faut l'intervention humaine car les échantillons sont segmentés manuellement.

### **1.4.3 Particules dynamiques**

Avec les modèles déformables basés sur les ensembles de niveaux, les particules dynamiques constituent une autre approche de segmentation pour une topologie arbitraire des objets. Présentée par Szeliski, Tonnesen et Terzopoulos (1993), cette méthode utilise un système dynamique de particules orientées, qui découvre la structure topologique et géométrique de la surface de l'objet selon la dynamique moléculaire. Deux types de forces sont en jeu : des forces d'attraction de longue portée et des forces de répulsion de courte portée. Pour appliquer ce modèle à la segmentation 3D de volume, les auteurs utilisent d'abord un opérateur 3D de détection de contours (voir la section 1.4.1) et emploient les contours pour initialiser et attirer les particules, ou utilisent directement les gradients des images 3D comme forces externes sur les particules. Les



particules sont ensuite reliées pour former une surface continue avec une triangulation de Delaunay<sup>2</sup>.

Szeliski, Tonnesen et Terzopoulos ont appliqué ce modèle de particules dynamiques pour la segmentation 3D à partir d'images médicales de différentes modalités, mais également pour les problèmes de correspondance de surfaces et de création de surfaces à partir de silhouettes. L'interaction des particules entre elles est coûteuse en terme de complexité de calcul, mais peut être réduite en considérant seulement un voisinage restreint pour chaque particule et en ne recalculant les nouvelles valeurs d'attraction et de répulsion qu'occasionnellement.

#### **1.4.4 Propagation d'ondes et traçage arrière**

Il est possible d'utiliser les notions de propagation d'ondes à travers un médium pour segmenter des images médicales sans souci topologique, le médium étant fourni par les images elles-mêmes. Kirbas et Quek (2003) ont développé une approche pour extraire le système vasculaire à partir d'images médicales 3D utilisant la propagation d'ondes et le traçage arrière.

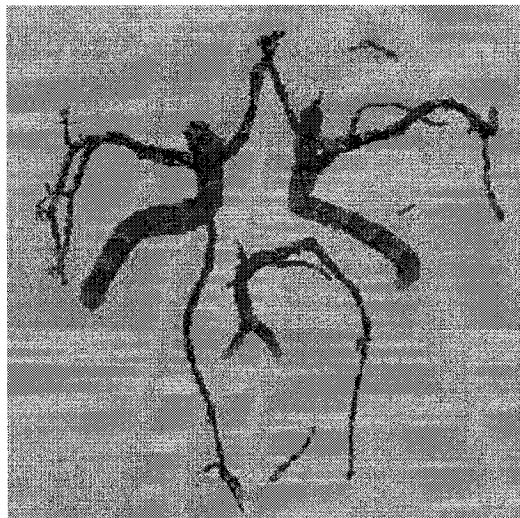
Dans un premier temps, un filtre est appliqué sur les images, assignant à chaque pixel une probabilité de faire partie d'un vaisseau sanguin en se basant sur l'intensité. Le problème est par la suite reformulé de cette façon : les probabilités sont considérées comme des indices de réfraction et l'image comme un médium à travers lequel une onde peut être propagée. Un voxel ne pouvant pas faire partie du système vasculaire a un indice infini, bloquant l'onde, un voxel en faisant certainement partie a un indice de 1, et les autres ont un indice assigné de façon linéaire entre 1 et le plus haut indice. De cette façon, si une onde est propagée à partir de la base d'un système vasculaire, elle voyagera

---

<sup>2</sup> Le lecteur intéressé à la triangulation de Delaunay est invité à consulter les livres de George et Borouchaki (1997) ou Frey et George (1999).

plus vite à travers les voxels ayant plus de chance de faire partie du système et inversement.

Puisque les images sont des données discrètes, l'algorithme de propagation de l'onde utilise une forme d'onde discrète en dents de scie. Le traçage arrière, qui retrouve le chemin parcouru par l'onde, s'effectue le long de la direction des normales locales du front de l'onde. La figure 1.19 montre la propagation et le traçage arrière pour des données provenant d'ARM.



**Figure 1.19 : Segmentation 3D de données provenant d'angiographie par résonance magnétique par propagation d'une onde et traçage arrière. (Kirbas et Quek 2003)**

Selon les auteurs, cette approche permet d'extraire non seulement les vaisseaux individuellement, mais également la connectivité et des informations morphologiques. Le système est robuste au bruit, facile à implanter et rapide, peut se propager à travers des formes arbitraires sans connaissance a priori sur la topologie. La constitution du filtre assignant à chaque pixel une probabilité de faire partie du système vasculaire pose toutefois une difficulté non négligeable.

## 1.5 Lissage de surfaces

Dans les sections précédentes, nous avons vu plusieurs méthodes et algorithmes pour extraire les données médicales et reconstituer un modèle tridimensionnel à partir de ces données, soit par une segmentation image par image suivie d'une reconstruction ou soit par une segmentation sur l'ensemble des images. Dans le cadre de la modélisation géométrique d'un organe pour des fins médicales, le modèle résultant doit être le plus fidèle possible à la réalité. Son apparence doit donc être lisse, c'est-à-dire exempte d'oscillations non désirables assimilables à du bruit.

Certaines méthodes de segmentation effectuent un lissage comme « effet secondaire », tel que le modèle de ballon. Cela est dû aux propriétés intrinsèques du modèle (par exemple ses contraintes de rigidité), qui lui confèrent un effet régulateur. Mais dans le cas de plusieurs méthodes de segmentation ou de reconstruction, notamment les particules dynamiques et la fragmentation directe, il est nécessaire d'effectuer une étape de lissage séparément. Et dans la plupart des travaux sur la modélisation d'organes à partir d'images médicales utilisant les méthodes de segmentation et de reconstruction présentées dans ce chapitre, les chercheurs sont exposés au problème de l'amélioration de surfaces tridimensionnelles car la dégradation des données sources peut introduire des erreurs. L'une des difficultés liées au lissage de surfaces est sans doute la conservation de la géométrie globale et du volume de l'objet. En effet, les techniques traditionnelles de lissage ont tendance à rétrécir l'objet lissé car à la limite, la surface obtenue converge en un point. Les paragraphes qui suivent visent à introduire brièvement le lecteur à quelques approches de lissage de surfaces. Une revue plus complète de la littérature à ce sujet serait impertinente dans le contexte du présent projet.

Beaucoup d'idées dans le traitement de surfaces 3D proviennent du traitement de signal. La généralisation des outils du domaine du traitement de signal et d'images aux surfaces 3D est devenue une problématique importante en infographie, particulièrement en

visualisation. Comme le lissage a été étudié de façon approfondie dans le domaine du traitement d'images, plusieurs chercheurs ont tenté d'étendre ces méthodes linéaires et non linéaires au lissage de surfaces. Les travaux de Taubin, Zhang et Golub (1996) sur la constitution de filtres passe-bas visant à préserver le volume de l'objet sont de bons exemples.

Deux tendances majeures dans le domaine du lissage de surfaces concernent le filtrage des sommets, par opposition au filtrage des vecteurs normaux à la surface. Dans la première catégorie, mentionnons le lissage laplacien, qui opère selon l'assomption que le bruit a une haute fréquence spatiale. Ce bruit peut être atténué par des opérations locales qui rendent les positions des sommets plus cohérentes avec leur voisinage. Cet opérateur a comme effet indésirable de réduire le volume de l'objet. Mentionnons également les travaux qui visent à améliorer le lissage laplacien, comme ceux de Taubin (1995a, 1995b) et de Vollmer, Mencl et Muller (1999), qui déplacent à nouveau les sommets filtrés par un processus d'inflation, et ceux de Liu, Bao, Heng, Wong et Peng (2001) et de Liu, Bao, Shum et Peng (2002), qui ajoutent des contraintes additionnelles aux sommets pour conserver le volume. Desbruns, Meyer, Schröder et Barr (1999) ont plutôt utilisé le principe du flux de courbure pour modifier les positions des sommets pour éviter le problème de dérivation tangentielle des sommets introduit par les méthodes basées sur la diffusion. Fleishman, Drori et Cohen-Or (2003) filtrent les sommets d'une manière basée sur le filtre bilatéral du traitement d'images, pour éviter également les problèmes de dérivation tangentielle des sommets. Dans la seconde catégorie, les auteurs favorisent le lissage des vecteurs normaux au lissage des sommets des polygones de la surface. Une telle approche requiert l'étape subséquente de l'adaptation de la surface aux nouvelles normales. Les travaux de Taubin (2001), Ohtake, Belyaev et Seidel (2002), Yagou, Ohtake et Belyaev (2002), Tasdizen, Whitaker, Burchard et Osher (2002, 2003), Mashiko, Yagou, Wei, Ding et Wu (2004) et ceux de Shen et Barner (2004) suivent cette idée.

À la lumière des travaux qui ont été revus dans ce chapitre, la section suivante précise les objectifs du projet et l'hypothèse de travail qui a été retenue.

## **1.6 Objectifs et hypothèse du mémoire**

L'objectif général du projet est de structurer une approche logicielle pour la modélisation de surfaces tridimensionnelles à partir de données tridimensionnelles. L'application principalement utilisée dans ce mémoire consiste à modéliser des vaisseaux sanguins à partir d'images médicales. Il s'agit donc de reconstruire la surface tridimensionnelle de cet organe en se basant sur les données physiques et géométriques réelles d'un patient telles que fournies par les images médicales obtenues via imagerie par résonance magnétique<sup>3</sup>. Les paragraphes qui suivent visent à introduire les éléments menant à l'élaboration de l'hypothèse de travail.

La revue bibliographique suggère que dans bon nombre d'approches, les auteurs créent un modèle polygonal de la surface en premier lieu (soit directement à partir du volume de données ou à partir de contours 2D préalablement trouvés image par image) ou après quelques traitements élémentaires sur les images médicales, puis tentent d'améliorer la fidélité du modèle reconstruit par des opérations sur les polygones. Tel est le cas pour les méthodes utilisant les modèles déformables. Dans ce mémoire, nous explorons une piste quelque peu différente en tentant d'améliorer la fidélité du modèle par le traitement des données sources. La création de la surface polygonale finale est réalisée en dernière étape du processus. Nous travaillons donc sur les intensités des voxels tout en demeurant conscients que ces valeurs serviront directement à produire la surface finale. En effet, nous considérons la surface finale en termes d'isosurface et d'isovaleur. Les

---

<sup>3</sup> Dans le cas de l'application principale, nous considérons que les images médicales constituant le volume de données sont alignées les unes par rapport aux autres. Le problème de recalage est en effet un sujet d'étude en soi.

caractéristiques de la surface sont donc implicitement contenues dans les valeurs mêmes du bain de voxels.

Dans le présent chapitre, nous avons eu l'occasion de présenter brièvement certaines des sous-étapes de la méthode de Cebal et Löhner (2001) pour la création de maillages d'éléments finis utilisables en simulation d'écoulement sanguin à partir d'images médicales, soient la croissance de région (section 1.2.3) et la fragmentation directe (section 1.3.3). La méthode complète se divise en trois étapes principales : 1) le traitement des images afin d'obtenir la frontière correctement définie de la paroi artérielle dans la région d'intérêt des images (segmentation directe); 2) la modélisation géométrique, qui rend la surface issue de la première étape par fragmentation directe topologiquement cohérente, lisse la surface et améliore la qualité de la triangulation; 3) la génération de maillage pour des fins de simulations. Notre méthode proposée se rapproche de celle des auteurs, qui accordent beaucoup d'importance au traitement 3D des données sources. Toutefois, Cebal et Löhner passent par un processus de binarisation des voxels, c'est-à-dire qu'après la croissance de région, les voxels peuvent prendre uniquement deux valeurs : soit qu'ils font partie de l'organe à modéliser (1) ou non (0). Ils enchaînent avec d'autres traitements sur les voxels binarisés avant de créer la surface. Dans notre projet, contrairement à Cebal et Löhner, les valeurs possibles des voxels seront toujours continues entre 0 et 1, afin de permettre une plus grande latitude au niveau des traitements appliqués sur les voxels.

Les modèles déformables constituent probablement la classe de méthodes de segmentation d'images médicales la plus courante. Il en existe plusieurs variantes, dont les principales ont été présentées à la section 1.4.2. L'une des caractéristiques les plus importantes, commune aux modèles déformables les plus usuels comme la surface active, le ballon et la représentation à l'aide des ensembles de niveaux, est certainement l'effet régulateur qui tend à lisser le modèle. Les contraintes internes du modèle, telles que la tension et la rigidité, permettent d'éliminer certains artéfacts des images médicales en

empêchant le modèle de « s'échapper » dans des zones non significatives (par exemple dans de faux débuts de protubérances) et en le forçant à dépasser des trous inexistant dans la réalité. Dans notre projet, nous avançons qu'il est possible de reproduire un effet semblable en appliquant des traitements sur les voxels.

Notre approche se veut uniformisante : en travaillant directement en trois dimensions sur les valeurs continues entre 0 et 1 des voxels, nous cherchons à pouvoir reproduire les effets de la méthode de segmentation directe et des modèles déformables, et donc leurs avantages, et ce avec des opérations relativement simples. Appliquer des traitements directement sur les valeurs des voxels semble en effet une opération simple en comparaison aux modèles physiques mécaniques impliqués dans les modèles déformables. Travailler dans le format des données sources permet de conserver un lien direct avec les données sources. De plus, le fait de considérer les trois dimensions des données permet d'utiliser facilement les informations reliées à la cohérence spatiale des données.

Enfin, l'approche cherche à retrouver un modèle géométrique réaliste. Dans le cas des images médicales, cela implique la production d'une surface qui est en accord avec l'anatomie humaine.

À la suite de ces considérations, l'hypothèse de travail suivante est posée :

- Travailler directement sur les valeurs des données tridimensionnelles, c'est-à-dire conserver le format « bain de voxels », constitue une approche uniformisante de la méthode de segmentation directe et des modèles déformables 3D tel que le ballon, permettant de profiter de la cohérence des données sources et de retrouver un modèle géométrique réaliste par des opérations simples.

Afin de vérifier l'hypothèse de travail, des objectifs spécifiques sont définis :

1. Développer un cadre de travail permettant de tester plusieurs algorithmes et facilitant l'ajout de nouvelles fonctionnalités ou méthodes alternatives; favoriser la réutilisation de design et de code pour produire un outil employable dans d'autres contextes d'application.
2. Implanter sous la forme de filtres des traitements unitaires applicables sur des bains de voxels.
3. Proposer des séquences de traitements permettant de retrouver une surface tridimensionnelle pour quelques groupes de données sources types.
4. Valider qualitativement et quantitativement le système de modélisation d'une surface à l'aide de modèles connus (analytiques) et réels.

Le premier objectif spécifique a trait directement à la conception logicielle du projet. Il fait intervenir des principes du domaine du génie logiciel tels que les patrons de conception. Évidemment, le présent mémoire n'a pas la prétention de trouver la meilleure combinaison de traitements possible qui fonctionne avec toutes les données tridimensionnelles. Le but de développer un outil de travail qui facilite l'ajout de nouvelles fonctionnalités va en ce sens. L'exemple d'application avec les images médicales a été choisi en partie à cause qu'il représente un grand défi. Toutefois, la modélisation d'une surface 3D à partir d'un volume de données ne se limite pas au domaine médical, elle est impliquée dans un bon nombre d'autres disciplines scientifiques<sup>4</sup>. La deuxième partie de cet objectif spécifique cherche à tenir compte de cette observation.

Dans le second objectif, il est question de diviser l'approche en traitements unitaires. Il s'agit donc de choisir et d'implanter des opérations agissant sur les intensités des voxels en 3D, chaque opération ayant son utilité spécifique.

---

<sup>4</sup> La météorologie en est un exemple. À partir des mesures de température prises en 3D dans l'atmosphère, il peut être intéressant de modéliser la surface des zones où la température atteint une certaine valeur critique.



Le troisième objectif spécifique constitue le cœur du projet. Il suggère qu'il est possible de combiner des traitements unitaires sur les données sources de manière à produire la surface tridimensionnelle désirée. La combinaison des traitements peut varier selon les données sources, c'est pourquoi quelques exemples sont considérés. La surface produite sera rendue disponible pour des étapes subséquentes de projets de recherche, par exemple pour des simulations numériques, dans quelques formats de fichiers standards.

La validation mentionnée dans le quatrième objectif spécifique se fera qualitativement de façon visuelle et quantitativement à l'aide de quelques métriques, c'est-à-dire l'aire de la surface du modèle et le volume compris à l'intérieur de la surface. L'utilisation de modèles analytiques offre la connaissance a priori des valeurs métriques de validation, permettant la comparaison avec les résultats produits par le système.

-----

Dans ce chapitre, nous avons fait la revue de l'état de l'avancement des travaux dans le domaine de la segmentation et de la modélisation d'organes à partir d'images médicales en séparant les méthodes des auteurs en deux approches : 1) segmentation 2D suivie d'une reconstruction 3D, et 2) segmentation 3D. Nous avons par la suite précisé les objectifs du projet et situé l'hypothèse de travail parmi les travaux de recherche du domaine.

Le chapitre suivant présente le cœur de la méthodologie choisie pour atteindre les objectifs et vérifier l'hypothèse, c'est-à-dire les traitements à appliquer sur les données sources pour obtenir des modèles 3D des vaisseaux sanguins.

## CHAPITRE 2 : TRAITEMENTS SUR LES DONNÉES SOURCES

Ce chapitre décrit les diverses opérations retenues pour le traitement tridimensionnel des données sources dans le but de retrouver un modèle surfacique 3D. Les traitements présentés agissent en 3D mais pourraient être développés en 4D selon les mêmes principes. L'amélioration de la qualité des données sources (la diminution des dégradations présentes) est primordiale et facilite grandement le processus de segmentation. L'application de ces traitements vise à améliorer la qualité des données sources puis à sélectionner l'objet que l'on cherche à modéliser (voir la section 1.2). Les traitements retenus sont : l'augmentation du contraste, le filtre de moyenne pondérée, le filtre médian, les opérations morphologiques et la croissance de région. Les mots « traitement » et « opération » seront utilisés de façon interchangeable dans le reste du document.

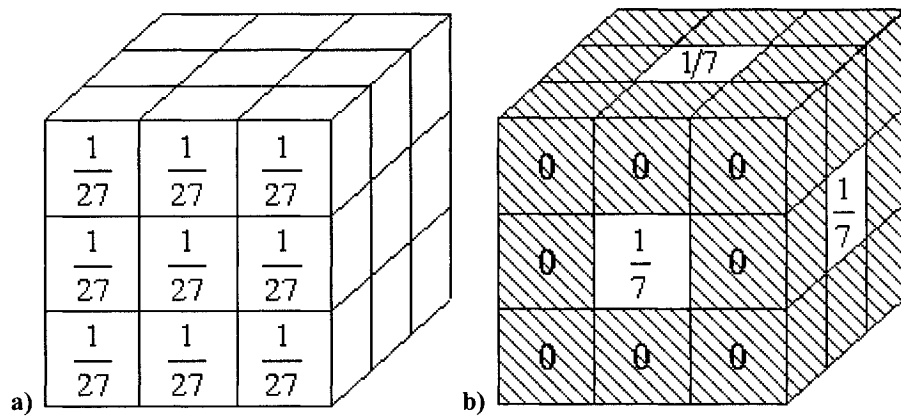
Les traitements présentés peuvent être considérés comme des filtres, c'est-à-dire des opérateurs qui transforment une série de données tridimensionnelles en une autre série de données tridimensionnelles selon une fonction de transfert qui leur est propre. La formule suivante résume cette idée :

$$I_{finale}(i, j, k) = f(I_{initiale}(i, j, k)), \quad (2.1)$$

où  $I_{initiale}(i, j, k)$  représente l'intensité initiale avant l'application du filtre et  $I_{finale}(i, j, k)$  l'intensité finale après l'application du filtre du voxel situé à l'emplacement correspondant aux indices  $i, j$  et  $k$ ;  $f$  indique que  $I_{finale}$  est fonction de  $I_{initiale}$ .

Ces opérations s'exécutent sur le domaine spatial des données sources. Dans le cas discret, les données sources sont représentées sous forme de voxels, et le filtre sous forme de noyau. Les filtres discutés dans ce chapitre sont basés sur le principe de voisinage : la valeur d'un voxel est calculée à partir des valeurs des voxels voisins. Le noyau sert à définir à la fois ce voisinage et en partie la fonction de transfert du filtre. Un

noyau est une fenêtre tridimensionnelle de petite taille, généralement de  $3 \times 3 \times 3$  ou  $5 \times 5 \times 5$  voxels, dont les voxels ont des valeurs assignées de façon à contrôler le voisinage et la fonction de transfert. La nouvelle valeur d'un voxel source est déterminée en superposant la fenêtre aux données sources. Cette fenêtre agit localement et est « glissée » sur l'ensemble des données sources. Le voxel qui se trouve au centre du noyau est celui pour lequel une nouvelle valeur est déterminée. La figure 2.1 donne deux exemples de noyaux de taille  $3 \times 3 \times 3$ .



**Figure 2.1 : Exemples de noyaux pour l'application de filtres sur des données tridimensionnelles. En a), tous les voxels voisins connectés au voxel central par au moins un coin commun participent de manière égale à la nouvelle valeur du voxel central (filtre de moyenne arithmétique). En b), seulement les voxels connectés au voxel central par une face partagée sont considérés. Les positions ayant une valeur nulle (voxels hachurés) ne sont pas considérées.**

Pour une partie des traitements, les données subissent une convolution avec le noyau. Cela se traduit par une somme de multiplications selon la formule suivante, pour un noyau de  $3 \times 3 \times 3$  :

$$I_{finale}(i, j, k) = \sum_{l=-1}^1 \sum_{m=-1}^1 \sum_{n=-1}^1 h(l, m, n) * I_{initiale}(i-l, j-m, k-n), \quad (2.2)$$

où  $h(l, m, n)$  représente la valeur du noyau à la position donnée par les indices  $l$ ,  $m$  et  $n$ . L'augmentation du contraste et le filtre de moyenne pondérée en sont des exemples. Dans les autres cas, par exemple pour un filtre non linéaire comme le filtre médian ou les opérations morphologiques, l'opération effectuée entre les données et le noyau n'est

pas une convolution. C'est pourquoi le noyau seul ne suffit pas à déterminer la fonction de transfert d'un filtre.

Le vocabulaire concernant les traitements étant établi, voyons maintenant les différents traitements retenus un peu plus en détails.

## 2.1 Augmentation du contraste

Le contraste réfère à la luminosité (ou obscurité) globale des objets sur une image (Bose 2004, p.11) ou, dans notre cas, sur une série de données tridimensionnelles. Lorsque le contraste est pauvre, il est difficile de différencier les objets entre eux. Traditionnellement, l'augmentation du contraste est basée sur l'histogramme des données. L'histogramme est un diagramme à barres représentant la distribution de fréquences d'une variable. Une définition plus spécifique, tirée de Nikolaidis et Pitas (2001, p.37), est qu'un histogramme est la fonction de densité de probabilité empirique des intensités des voxels. Pour des données tridimensionnelles de dimensions  $N_i \times N_j \times N_k$  pouvant prendre  $M$  différentes valeurs discrètes d'intensités  $i_1, \dots, i_M$  (par exemple de 0 à 255), la valeur de l'histogramme à une intensité donnée est la fréquence relative d'occurrence de cette intensité :

$$p(i_M) = \frac{n_M}{N_i N_j N_k} \quad (2.3)$$

où  $n_M$  est le nombre de voxels dont l'intensité est  $i_M$ .

Un pauvre contraste se traduit à travers l'histogramme par une concentration des fréquences d'occurrence dans une plage réduite d'intensités, située entre deux seuils. L'augmentation du contraste cherche à étirer cette plage d'intensités en éloignant les seuils minimal et maximal l'un de l'autre.

Dans ce projet, nous avons retenu une approche quelque peu différente d'augmentation du contraste. Au lieu de calculer l'histogramme des données, les intensités des voxels sont modifiées selon des seuils spécifiés par l'utilisateur. En effet, l'histogramme d'une image médicale contient rarement une seule zone de concentration des fréquences d'occurrence bien définie. En utilisant la formule suivante (Fisher, Perkins, Walker et Wolfart 2004) :

$$I_{finale}(i, j, k) = [I_{initiale}(i, j, k) - Seuil_{min}] * \frac{I_{max}}{Seuil_{max} - Seuil_{min}} \quad (2.4)$$

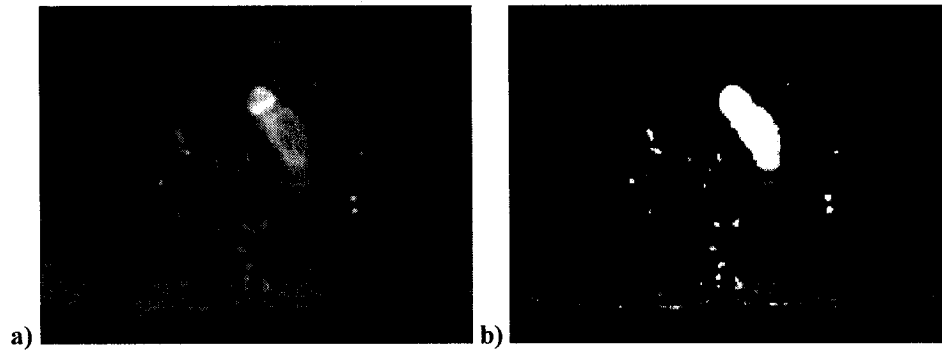
où  $I_{max}$  est la valeur maximale possible pour l'intensité (généralement de 1,0),  $Seuil_{min}$  et  $Seuil_{max}$  représentent les valeurs seuils, et  $I_{finale}(i,j,k)$  est restreinte entre 0 et 1, et en choisissant judicieusement ces valeurs seuils, il est possible d'éliminer une intéressante partie des artéfacts présents sur les données, de même que certains organes qu'on ne cherche pas à modéliser et qui présentent des valeurs d'intensité situées à l'extérieur des seuils. Cette formule permet de travailler avec des valeurs décimales entre 0 et 1.

En termes de filtre, cette augmentation du contraste peut être assimilable à une convolution<sup>5</sup> entre les données sources et le noyau, ce dernier étant de dimensions 1 x 1 x 1. Chaque voxel sera donc modifié en fonction de son intensité initiale, sans tenir compte de ses voisins, selon la valeur du noyau. L'algorithme utilisé est présenté à l'annexe B sous forme de pseudocode. Sa complexité est de l'ordre de  $O(N)$ , où  $N$  est le nombre de voxels des données.

La figure 2.2 illustre le résultat de l'application du filtre d'augmentation du contraste sur une série d'images médicales 3D, avec des seuils minimal et maximal spécifiés à 0,3 et 0,4, respectivement. Un plan image est montré.

---

<sup>5</sup> Ce n'est pas exactement une convolution puisque la valeur initiale subit une soustraction avant d'être multipliée.



**Figure 2.2 : Application du filtre d'augmentation du contraste avec des seuils minimal et maximal de 0,3 et 0,4. Données initiales en a) et données résultantes en b).**

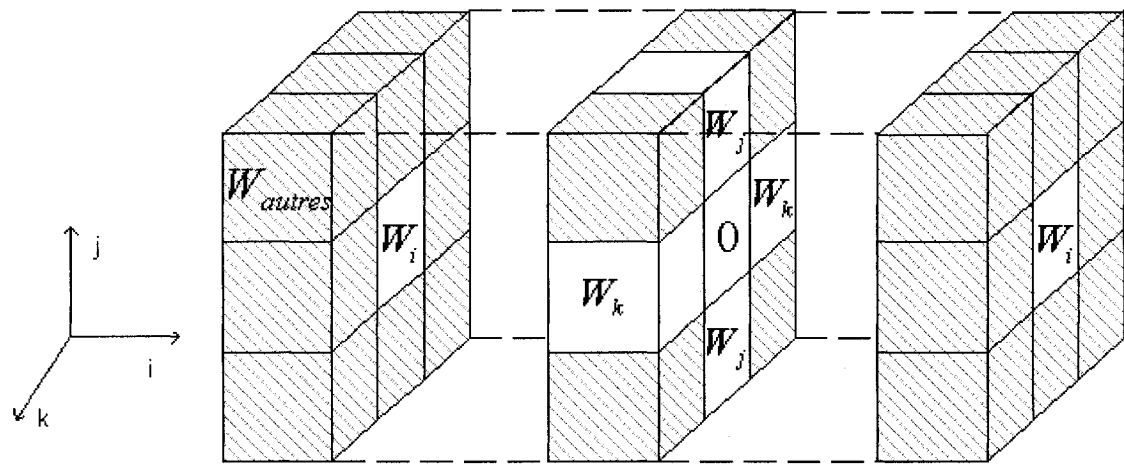
## 2.2 Filtre de moyenne pondérée

Le filtre de moyenne pondérée est un opérateur d'adoucissement des données. Le but du filtre est de réduire la quantité de variation entre les voxels voisins et ainsi de rendre les intensités des voxels plus cohérentes localement. Cette opération a comme effet parfois indésirable de rendre les contours des objets plus flous.

Le filtre de moyenne est une opération bien connue du domaine du traitement d'images. Les données sources subissent une convolution avec un noyau. Ce noyau permet d'effectuer la moyenne arithmétique des intensités du voxel central et des voxels voisins et d'affecter cette moyenne comme la nouvelle valeur du voxel central. En trois dimensions, le noyau ressemble à celui présenté à la figure 2.1a.

Dans le cadre du présent projet, nous avons choisi d'utiliser une version plus générale du filtre de moyenne. Les voisins du voxel central n'ont pas nécessairement le même poids, d'où le nom « filtre de moyenne pondérée ». Ces poids sont divisés en quatre catégories : poids des voxels immédiatement voisins en  $i$  ( $W_i$ ), poids des voxels immédiatement voisins en  $j$  ( $W_j$ ), poids des voxels immédiatement voisins en  $k$  ( $W_k$ ) et poids des autres voxels voisins situés dans la fenêtre du noyau ( $W_{\text{autres}}$ ). En affectant une valeur égale à tous les poids, on retrouve la formule de la moyenne arithmétique, à

l'exception que le voxel central ne contribue pas à sa nouvelle valeur. Le pseudocode de l'algorithme utilisé est présenté à l'annexe B. La complexité est de l'ordre de  $O(N K^3)$ , où  $N$  est le nombre de voxels des données et  $K$  le côté de la fenêtre cubique du noyau. Le coût augmente donc rapidement avec la taille du noyau, c'est pourquoi un côté de  $K = 3$  est généralement utilisé. La figure 2.3 présente le noyau plus général du filtre de moyenne pondérée.

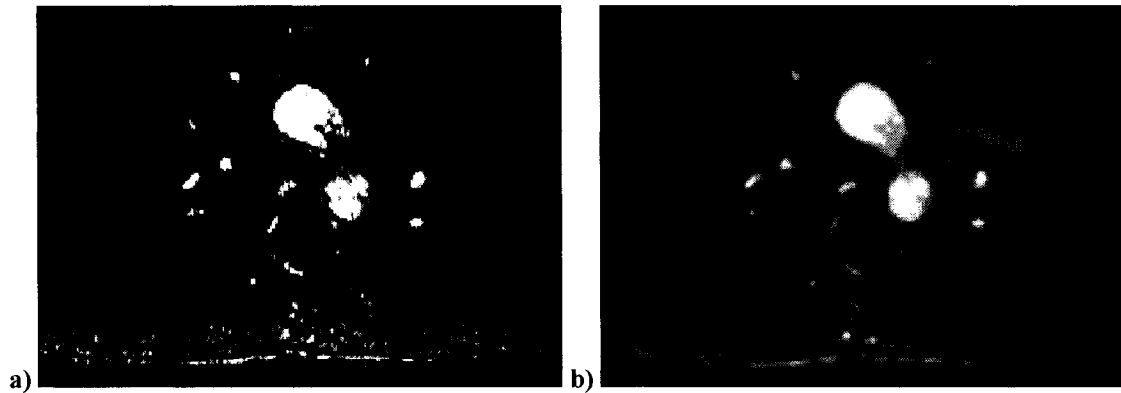


**Figure 2.3 : Noyau du filtre de moyenne pondérée.** Les plans j-k ont été éloignés les uns des autres afin de voir le centre du noyau. Différents poids sont associés aux voxels du filtre selon la position du voxel : les deux voxels immédiatement voisins en i du voxel central (de poids 0) ont un poids  $W_i$ , les deux voxels immédiatement voisins en j du voxel central ont un poids  $W_j$ , les deux voxels immédiatement voisins en k du voxel central ont un poids  $W_k$ , les 20 autres (hachurés) ont un poids  $W_{\text{autres}}$ .

L'intérêt de cette formulation est de pouvoir spécifier un filtre anisotrope, c'est-à-dire un filtre dont les valeurs varient selon la direction considérée. Pour le traitement d'images médicales, cet aspect devient intéressant car le volume de données créé par l'empilement des images médicales est souvent lui-même anisotrope, la taille d'un voxel étant différente dans la direction k, c'est-à-dire la direction du déplacement du scanneur.

La figure 2.4 donne un exemple de l'application du filtre de moyenne pondérée sur une série d'images médicales 3D, avec des poids  $W_i$  et  $W_j$  de  $2/34$ ,  $W_k$  de  $3/34$  et  $W_{\text{autres}}$  de  $1/34$ , où 34 représente la somme totale des numérateurs des poids des 26 voxels faisant partie du noyau. Un plan image est montré. Puisque l'application du filtre se fait en trois

dimensions, les images précédant et suivant l'image 2D montrée influencent les nouvelles intensités, ce qui fait apparaître des structures absentes sur l'image des données initiales (voir la flèche).



**Figure 2.4 :** Application du filtre de moyenne pondérée, avec des poids  $W_i$  et  $W_j$  de  $2/34$ ,  $W_k$  de  $3/34$  et  $W_{\text{autres}}$  de  $1/34$ . Données initiales en a) et données résultantes en b).

### 2.3 Filtre médian

Le filtre médian, à l'instar du filtre de moyenne pondérée, constitue un opérateur d'adoucissement de données. Toutefois, il vise à éliminer le bruit impulsif et tend à préserver les contours des objets relativement bien (Nikolaidis et Pitas 2001). Ce filtre fait partie de la famille des filtres de rang car il est basé sur l'ordonnancement de valeurs. Il n'y a donc pas d'opération de convolution, contrairement aux deux traitements précédents. Le noyau du filtre sert uniquement à spécifier le voisinage à considérer : les voxels des données sources correspondant à des valeurs de 0 dans le noyau ne feront pas partie du calcul.

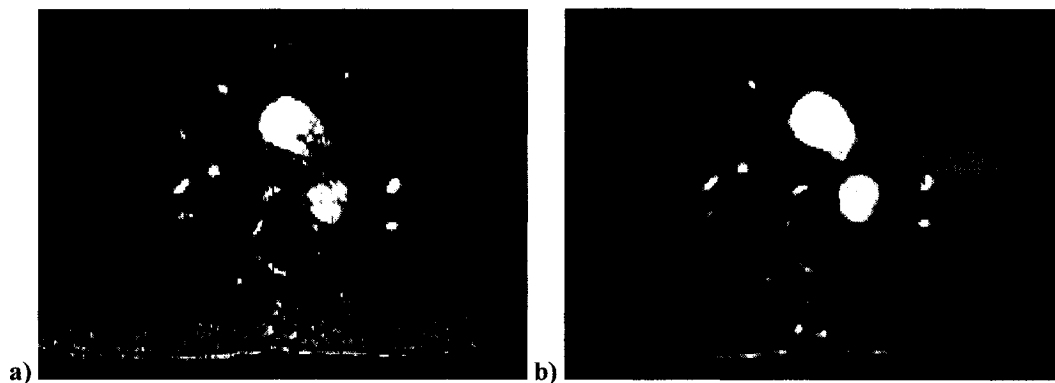
Tel que son nom le laisse entrevoir, le filtre médian choisit la valeur médiane parmi les voxels voisins considérés comme nouvelle valeur du voxel central. Pour une série de données classées par amplitude croissante, la valeur médiane correspond à la valeur située au milieu de la série. L'algorithme de tri rapide (voir par exemple le chapitre 7 du livre de Cormen, Leiserson, Rivest et Stein 2001) peut être utilisé pour le classement des



valeurs des voxels. L'algorithme utilisé est présenté à l'annexe B sous forme de pseudocode. Sa complexité est de l'ordre de  $O(N K^3 \log K^3)$ , où  $N$  est le nombre de voxels des données et  $K$  le côté de la fenêtre cubique du noyau.

Le fait d'attribuer à la série de données traitées des valeurs provenant de l'ensemble des valeurs initiales (bruitées) et donc de ne pas créer de nouvelles valeurs permet de filtrer des données portant une information symbolique, par exemple des données issues d'une étape de segmentation (Maître 2003a). Le filtre médian est également insensible aux valeurs extrêmes, ce qui lui procure un avantage sur le filtre de moyenne arithmétique lorsque les données comportent de tels valeurs associées à du bruit, au détriment d'une complexité un peu plus élevée.

La figure 2.5 montre l'effet de l'application du filtre médian de taille  $3 \times 3 \times 3$  sur une série d'images médicales 3D. On retrouve le même plan image et les mêmes données initiales qu'à la figure 2.4 afin de permettre la comparaison avec le filtre de moyenne pondérée. À l'instar du filtre de moyenne pondérée, puisque l'application du filtre médian se fait en trois dimensions, les images précédant et suivant l'image 2D montrée influencent les nouvelles intensités, ce qui fait apparaître des structures absentes sur l'image des données initiales. On constate toutefois que le pont reliant les deux parties de la crosse de l'aorte (voir la flèche) n'est pas apparu et que les transitions entre les objets de l'image sont plus nettes.



**Figure 2.5 : Application du filtre médian. Données initiales en a) et données résultantes en b).**

## 2.4 Opération morphologique

La morphologie mathématique constitue un cadre de travail pour l'analyse et le traitement d'images qui agit au niveau de la forme spatiale ou de la structure d'objets, initialement formulé pour le traitement des données binaires (Nikolaidis et Pitas 2001). Dans le cas binaire, les opérations morphologiques peuvent être décrites simplement en termes d'addition et de soustraction de pixels selon certaines règles, qui dépendent des valeurs des pixels voisins (Russ 1999). Ces opérations peuvent être étendues aux images en niveaux de gris ou, plus généralement, à des séries de données représentées par une seule valeur, comme c'est le cas des voxels des données sources du présent projet.

Les règles gérant les opérations morphologiques utilisent des comparaisons entre les valeurs des données sources et un modèle. Ce modèle de comparaison, appelé élément structurant, est assimilable à un noyau. L'élément structurant que nous avons choisi est le plus simple : un noyau de  $2 \times 2 \times 2$  ou de  $3 \times 3 \times 3$  ayant une valeur de 1 à chaque position possible. Nous offrons aussi la possibilité d'utiliser un élément structurant de  $3 \times 3 \times 3$  pour lequel les valeurs de 1 sont positionnées en forme de croix. Les opérations morphologiques, filtres non linéaires, ne sont pas des convolutions avec les données.

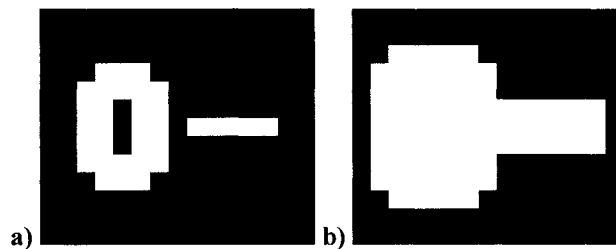
Les opérations les plus fréquentes sont la dilatation et l'érosion, et des combinaisons de celles-ci. Elles ont été retenues pour ce projet et sont présentées dans les sections suivantes.

### 2.4.1 Dilatation

Le processus de dilatation cherche à ajouter des voxels aux objets d'intérêt contenus dans les données (par exemple l'organe à modéliser dans les images médicales), en traitant les voxels classifiés comme ne faisant pas partie des objets mais ayant une connexion avec des voxels faisant partie des objets. Cette opération élargit les objets

présents dans les données sources, peut fusionner des objets situés très près les uns des autres et remplit les petits trous.

D'un point de vue ensembliste, la dilatation binaire impose une relation d'intersection entre l'objet dans les données et l'élément structurant : un point appartient à l'objet final si l'élément structurant centré en ce point intersecte l'objet initial (Bloch 2001). La figure 2.6 donne un exemple de dilatation binaire en deux dimensions par un élément structurant carré de 3 x 3 : l'objet de gauche, en blanc sur fond noir, s'en trouve élargi, son trou est rempli et il est fusionné avec l'objet de droite.



**Figure 2.6 : Dilatation binaire avec un élément structurant carré de 3 x 3 pixels. Image initiale en a) et image résultante en b). La taille de l'image est de 13 x 15 pixels.**

Le passage de la dilatation binaire à la dilatation fonctionnelle, nécessaire pour traiter des données médicales représentées par des niveaux de gris, peut se faire de plusieurs façons. Un choix simple est de remplacer les concepts ensemblistes par leurs équivalents fonctionnels en considérant la méthode des extrêmes : la nouvelle valeur d'un voxel correspondra à la valeur maximale située dans le voisinage spécifié par l'élément structurant (qui demeure binaire) (Bloch 2001, Pratt 2001). Cette opération peut être perçue comme un filtre de rang : tel que dans le cas du filtre médian, les valeurs du voisinage sont ordonnancées de façon croissante. Au lieu de choisir la valeur médiane, la dernière valeur (maximale) est retenue. Il est bien sûr possible, pour plus de simplicité, de ne pas trier les valeurs et de conserver la valeur maximale courante jusqu'à la fin de l'itération. L'algorithme utilisé est présenté à l'annexe B sous forme de pseudocode. Sa complexité est de l'ordre de  $O(N K^3)$ , où  $N$  est le nombre de voxels des données et  $K$  le côté de la fenêtre cubique de l'élément structurant.

La figure 2.7 illustre le résultat de l'application du filtre de dilatation fonctionnelle sur une série d'images médicales 3D une fois et deux fois, avec un élément structurant cubique de  $3 \times 3 \times 3$ . Un plan image est montré. On peut constater que sur des images 3D à niveaux de gris, la dilatation fonctionnelle propage les maxima locaux des niveaux de gris (Bloch 2001).

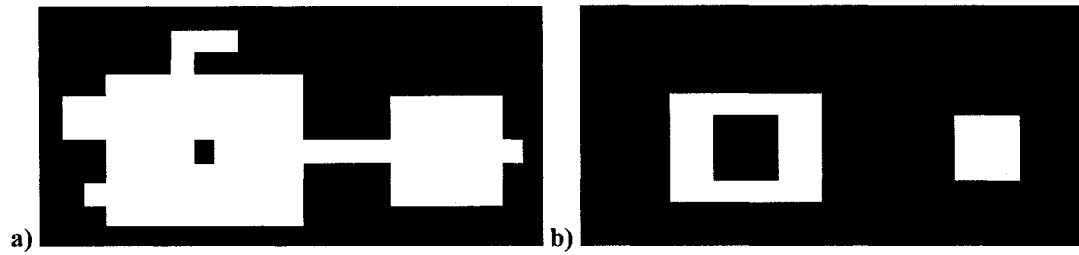


**Figure 2.7 : Application du filtre de dilatation fonctionnelle. Données initiales en a) et données résultantes après une application du filtre en b) et deux applications du filtre en c).**

### 2.4.2 Érosion

Le processus d'érosion est à l'opposé de la dilatation, sans être directement son inverse. Il cherche à soustraire des voxels aux objets d'intérêt contenus dans les données en traitant les voxels classifiés comme faisant partie des objets mais ayant une connexion avec des voxels ne faisant pas partie des objets. L'érosion rétrécit les objets présents dans les données sources, peut séparer complètement deux objets reliés par un isthme et élimine les petites protubérances.

D'un point de vue ensembliste, l'érosion binaire impose une relation d'inclusion entre l'objet dans les données et l'élément structurant : un point appartient à l'objet final si l'élément structurant centré en ce point est inclus dans l'objet initial. La figure 2.8 fournit un exemple d'érosion binaire en deux dimensions par un élément structurant carré de  $3 \times 3$  : l'objet initial, en blanc sur fond noir, est séparé en deux objets distincts, en plus d'être rétréci, de voir son trou agrandi et de perdre ses petites protubérances.



**Figure 2.8 : Érosion binaire avec un élément structurant carré de 3 x 3 pixels. Image initiale en a) et image résultante en b). La taille de l'image est de 23 x 11 pixels.**

Comme dans le cas de la dilatation, un choix simple pour passer de l'érosion binaire à l'érosion fonctionnelle est de remplacer les concepts ensemblistes par leurs équivalents fonctionnels en considérant la méthode des extrêmes. Pour l'érosion, la nouvelle valeur d'un voxel correspondra à la valeur minimale située dans le voisinage spécifié par l'élément structurant (qui demeure binaire). L'érosion fonctionnelle est également assimilable à un filtre de rang : il suffit de choisir la première valeur (minimale) parmi les valeurs ordonnées du voisinage. À l'instar de la dilatation fonctionnelle, il est bien sûr possible de ne pas trier les valeurs pour plus de simplicité. L'algorithme utilisé est présenté à l'annexe B sous forme de pseudocode. Il peut être combiné avec celui de la dilatation. Sa complexité est de l'ordre de  $O(N K^3)$ , où  $N$  est le nombre de voxels des données et  $K$  le côté de la fenêtre cubique de l'élément structurant.

La figure 2.9 montre un exemple de l'application du filtre d'érosion fonctionnelle sur une série d'images médicales 3D une fois et deux fois, avec un élément structurant cubique de 3 x 3 x 3. Un plan image est montré. On peut constater que sur des images 3D à niveaux de gris, l'érosion fonctionnelle propage les minima locaux des niveaux de gris (Bloch 2001).



**Figure 2.9 : Application du filtre d'érosion fonctionnelle. Données initiales en a) et données résultantes après une application du filtre en b) et deux applications du filtre en c).**

### 2.4.3 Ouverture / fermeture

Les opérations morphologiques de dilatation et d'érosion ne sont pas à l'inverse l'une de l'autre. Si l'on effectue une dilatation suivie d'une érosion, l'allure des données traitées sera différente que si l'on effectue une érosion suivie d'une dilatation. Un exemple simple suffit pour illustrer cette constatation : prenons, dans le cas à deux dimensions, une image en niveaux de gris comprenant différents objets d'intensités et de grosseurs variables, sur un fond obscur. Une première étape d'érosion peut faire disparaître complètement certains petits objets de taille similaire à la taille de l'élément structurant (voir figure 2.9b), qui ne pourront être récupérés par la seconde étape de dilatation et seront donc définitivement perdus. Une première étape de dilatation aura pour effet d'augmenter la taille de ces mêmes petits objets, qui ne pourront par la suite être éliminés par la seconde étape d'érosion et seront donc toujours présents dans l'image finale. Nous pouvons donc créer deux nouvelles opérations en combinant la dilatation et l'érosion : l'ouverture et la fermeture.

L'ouverture morphologique est accomplie en effectuant une érosion suivie d'une dilatation en utilisant le même élément structurant. Elle a pour effet, dans sa version fonctionnelle sur des images à niveaux de gris, d'éliminer les petits détails clairs (l'érosion obscurcit les données de façon générale, qui sont éclaircies à nouveau par la dilatation). La fermeture, à l'inverse de l'ouverture, élimine les petits détails foncés (la

dilatation rend les données plus claires de façon générale, puis l'érosion tend à restaurer la luminosité originale sans réintroduire les détails foncés) (Watt et Policarpo 1998).

Nous avons choisi de conserver un élément structurant binaire lors de la généralisation des opérations morphologiques en versions fonctionnelles. Il aurait été également possible d'utiliser des éléments structurants fonctionnels, mais nous avons opté pour la simplicité.

## **2.5 Croissance de région**

La croissance de région constitue une technique de segmentation. Sa principale utilité réside dans sa capacité à isoler un objet dans une série de données. Traditionnellement, à partir d'un voxel appelé germe, une région uniforme est grossie en y ajoutant progressivement les voxels voisins qui répondent au critère d'homogénéité (voir la section 1.2.3). Elle peut s'apparenter aux autres filtres présentés par son aspect de voisinage local considéré à chaque itération.

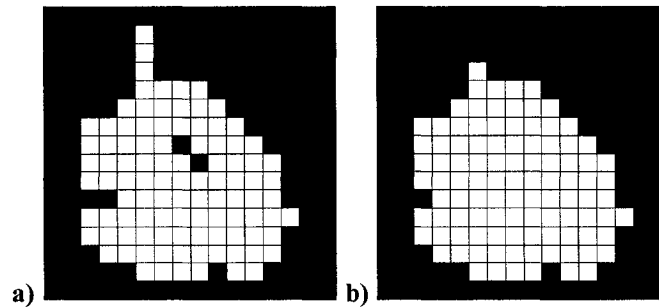
Dans le cadre de ce projet, nous avons développé une version de l'algorithme de croissance de région qui se comporte de façon similaire aux modèles déformables telle que le ballon quant à leur effet régulateur. À cause des contraintes intrinsèques du modèle déformable (rigidité, tension), la surface obtenue après segmentation est généralement lisse localement. Nous avons donc cherché à intégrer cette caractéristique dans la croissance de région en ajoutant au critère d'homogénéité traditionnel des contraintes produisant un effet semblable à celles des modèles déformables.

Comme nous l'avons vu aux sections 1.2.2.1 et 1.4.2.1 qui décrivaient le contour actif, la surface active et le ballon, les modèles déformables basés sur le contour actif répondent à trois catégories de forces, dont les deux principalement utilisées sont celles relatives à l'énergie interne et celles relatives à l'énergie de l'image. Les forces d'énergie interne

reposent généralement sur des principes de physique. Dans le modèle du ballon, une tension de surface est simulée en traitant les arêtes des polygones du modèle comme des ressorts, afin que le ballon conserve un aspect lisse en présence de bruit. Les forces d'énergie de l'image sont déterminées de façon à ce que le modèle soit attiré par les zones de changements importants d'intensité, qui coïncident la plupart du temps avec les limites des objets de l'image. Dans le cas de notre algorithme de croissance de région, l'élément correspondant à l'énergie de l'image demeure le critère d'homogénéité basé sur l'intensité des voxels : la région croît lorsqu'elle rencontre des voxels dont l'intensité fait partie de la bonne plage de valeurs (définie par des seuils minimal et maximal d'intensité, spécifiés par l'utilisateur), et s'arrête lorsque ses frontières touchent des voxels ayant une intensité à l'extérieur de la plage de valeurs, ce qui se produit la plupart du temps aux limites de l'objet. Cette croissance se fait en vérifiant si, parmi les six voxels immédiatement voisins (V) de tous les voxels de la région (voir la figure 1.7), certains doivent être intégrés à la région, à la manière de Cebal et Löhner (2001). Notre algorithme diffère des algorithmes de croissance de région traditionnels pour se rapprocher des modèles déformables par son intégration d'un élément correspondant à l'énergie interne à travers un filtre dans le but d'apporter un effet régulateur.

La régulation s'effectue en deux temps : afin d'obtenir une surface plus lisse, certains voxels doivent être inclus dans la région même si leur intensité ne se trouve pas dans la bonne plage, et certains voxels doivent être exclus de la région même si leur intensité se trouve dans la bonne plage et qu'ils sont connectés à la région. La figure 2.10 donne un exemple, pour une image 2D binaire, d'émulation du comportement d'un modèle déformable comme le ballon. Le modèle, de par ses contraintes intrinsèques, ne se déformerait pas dans les régions trop étroites et boucherait certains trous en passant outre à cause de la simulation d'une pression interne. L'arrière-plan est en noir.





**Figure 2.10 : Émulation de la segmentation d'une image binaire par un modèle déformable comme le ballon. Image originale en a) et image segmentée en b). La taille de l'image est de 16 x 16 pixels.**

Pour intégrer cette régulation à l'algorithme, nous avons choisi d'observer le voisinage des voxels voisins en traitement, compris dans une fenêtre de  $3 \times 3 \times 3$  voxels centrée sur le voxel voisin V. L'intensité d'un voxel peut être soit dans la bonne plage ou à l'extérieur de cette plage. Cette binarité facilite l'étude pour déterminer dans quels cas des voxels doivent être inclus ou exclus. En trois dimensions, 27 voxels à la fois font partie de la fenêtre considérée, ce qui constitue un total de  $2^{27}$  (134 217 728) configurations possibles à étudier (et non pas  $256^{27}$ , le nombre réel de possibilités avec les 256 niveaux d'intensité). Ce nombre est malgré tout beaucoup trop grand pour que toutes les possibilités soient étudiées dans le but de déterminer les règles d'inclusion et d'exclusion et de construire l'algorithme. Par contre, il est possible de faire l'étude en deux dimensions, pour une fenêtre de  $3 \times 3$ , et d'appliquer les résultats en trois dimensions. En effet, dans ce cas, 9 pixels à la fois font partie de la fenêtre, ce qui constitue un total de  $2^9$  (512) configurations possibles. Ce nombre peut être réduit à 144 en considérant que certaines configurations ne peuvent se présenter en pratique et en éliminant les symétries par rapport à la direction de la croissance de la région. Dans la fenêtre, le pixel situé au milieu de la rangée du bas est considéré comme le pixel faisant déjà partie de la région (pixel R, aux coordonnées  $i,j-1$ ). C'est son voisin immédiatement au-dessus de lui, le pixel central (pixel V, aux coordonnées  $i,j$ ), qui est en cours d'étude. La croissance de région provient donc du bas de la fenêtre, ce qui élimine les configurations où le pixel R serait à 0. La figure 2.11 illustre la fenêtre considérée et la position des pixels R et V.

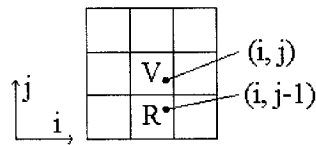


Figure 2.11 : Voisinage et position des pixels R et V pour la croissance de région.

Après avoir étudié les différentes configurations, nous avons retenu les cas permettant de tirer les règles à intégrer à l'algorithme de manière à simuler l'effet régulateur des modèles déformables comme le ballon. Ces cas sont illustrés aux deux prochaines figures. La figure 2.12 présente en a) les cas où il faut exclure le pixel V bien que son intensité soit acceptable (comprise entre les seuils minimal et maximal) et en b) les cas limites où il faut conserver le pixel V ayant une intensité acceptable. La figure 2.13 présente les cas où il faut inclure le pixel V bien que son intensité ne soit pas acceptable (plus petite que le seuil minimal ou plus grande que le seuil maximal). L'arrière-plan est en noir, le pixel d'où provient la croissance de région est indiqué par la lettre R, et le pixel voisin en traitement est indiqué par la lettre V.

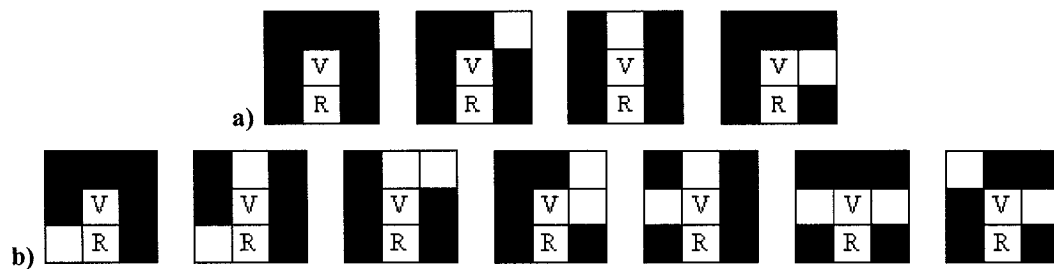
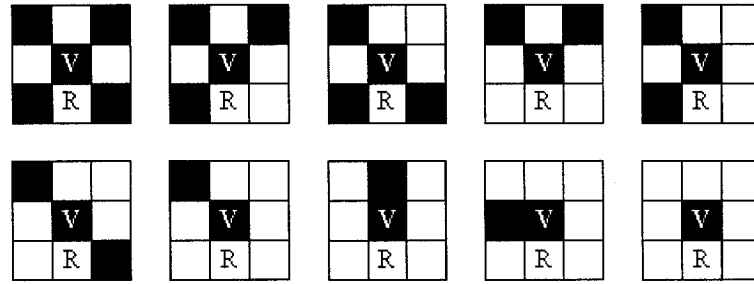


Figure 2.12 : Croissance de région avec pixel V d'intensité acceptable. En a), configurations pour lesquelles il faut exclure le pixel V et en b), configurations pour lesquelles il faut inclure le pixel V.



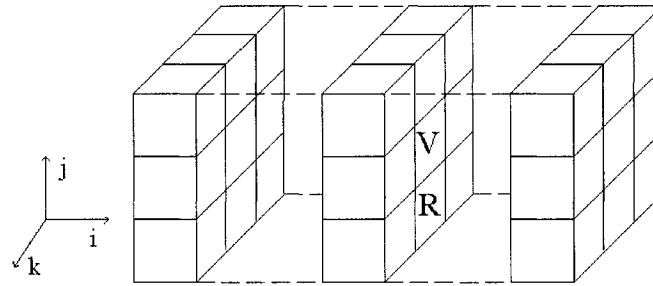
**Figure 2.13 : Croissance de région avec pixel V d'intensité non acceptable. Configurations pour lesquelles il faut inclure le pixel V.**

À la lumière des figures précédentes, nous avons tiré les règles suivantes en deux dimensions, les termes « intensité acceptable » signifiant que l'intensité est comprise entre les seuils minimal et maximal, et « intensité non acceptable » signifiant que l'intensité est plus petite que le seuil minimal ou plus grande que le seuil maximal :

- Si l'intensité du pixel V est acceptable,
  - S'il y a 3 pixels ou moins dans la fenêtre dont l'intensité est acceptable et que dans la rangée du côté de la provenance de la croissance (rangée  $i-1$ ), il n'y a que le pixel R ayant une intensité acceptable, le pixel V doit être exclu de la région.
  - Sinon, le pixel V doit être inclus dans la région.
- Si l'intensité du pixel V n'est pas acceptable,
  - S'il y a 7 pixels ou plus dans la fenêtre dont l'intensité est acceptable ou que les 4 pixels immédiatement voisins du pixel V (partageant un côté) ont une intensité acceptable, le pixel V doit être inclus dans la région.
  - Sinon, le pixel V doit être exclu de la région.

Le choix des nombres de pixels présents dans ces règles a été basé sur l'observation des cas présentés aux figures 2.12 et 2.13. Nous avons déterminé que ces valeurs particulières (3, 4 et 7) permettent de cataloguer les cas de la même façon qu'aux deux figures précédentes.

Les règles établies en deux dimensions peuvent être reformulées pour être appliquées en trois dimensions, en considérant une fenêtre de  $3 \times 3 \times 3$  comprenant 27 voxels. Les nombres du cas 2D, représentant des rapports sur 3 ou 9 pixels, sont simplement ramenés dans des rapports sur 9 ou 27 voxels pour le cas 3D. La croissance de région provient du voxel R, central au plan i-k du bas de la fenêtre, et le voxel sous étude est le voxel V, central à la fenêtre, tel qu'illustré à la figure 2.14.



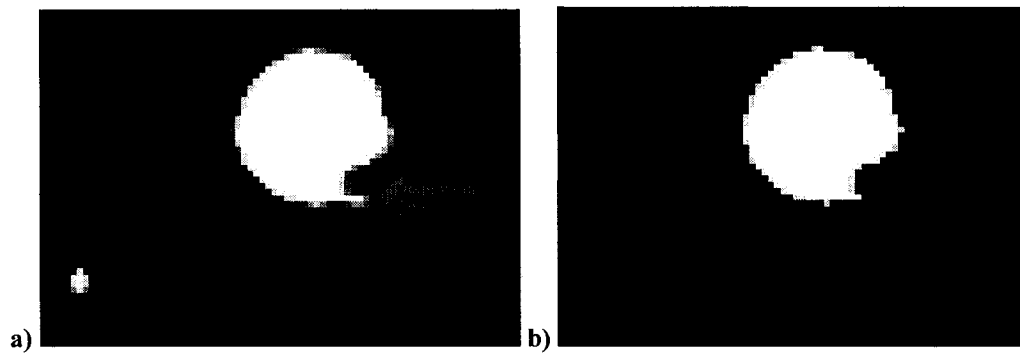
**Figure 2.14 : Croissance de région : illustration de la fenêtre  $3 \times 3 \times 3$  centrée sur le voxel immédiatement voisin en traitement (V). La croissance provient du voxel R.**

Les règles en 3D sont donc :

- Si l'intensité du voxel V est acceptable,
  - S'il y a 9 voxels ou moins dans la fenêtre dont l'intensité est acceptable et que dans le plan i-k du côté de la provenance de la croissance (celui en j-1), il n'y a que 3 voxels ou moins avec une intensité acceptable, le voxel V doit être exclu de la région.
  - Sinon, le voxel V doit être inclus dans la région.
- Si l'intensité du voxel V n'est pas acceptable,
  - S'il y a 21 voxels ou plus dans la fenêtre dont l'intensité est acceptable ou que les 6 voxels immédiatement voisins du voxel V (partageant une face) ont une intensité acceptable, le voxel V doit être inclus dans la région.
  - Sinon, le voxel V doit être exclu de la région.

L'algorithme utilisé est présenté à l'annexe B sous forme de pseudocode. Sa complexité, difficile à évaluer, dépend principalement du nombre de voxels inclus dans la région.

Selon les données sources, ce nombre représente une fraction du nombre de voxels total qui généralement tourne autour de 1 %. Si l'on considère une fraction de 10 %, la complexité de l'algorithme se situe entre les bornes  $O(N)$  et  $O(N K^3)$ , où  $N$  représente le nombre de voxels des données et  $K$  le côté de la fenêtre du voisinage considéré. La figure 2.15 montre un exemple d'application de la croissance de région sur une série d'images médicales 3D, avec des seuils minimal et maximal de 0,8 et 1,0. Un plan image est montré. Notons l'objet de gauche qui a été éliminé parce que la région n'a pu croître jusqu'à lui, et la protubérance (voir flèche) qui a été exclue.



**Figure 2.15 : Application de l'algorithme de croissance de région. Données initiales en a) et données résultantes en b).**

-----

Dans ce chapitre, nous avons introduit plusieurs traitements unitaires applicables sur des données sources tridimensionnelles : l'augmentation du contraste, le filtre de moyenne pondérée, le filtre médian, les opérations morphologiques et la croissance de région. Le choix de ces traitements, parmi toutes les opérations existantes possibles, s'est fondé sur leur simplicité et leur complémentarité en termes d'utilité. Le tableau 2.1 propose une récapitulation des caractéristiques principales des traitements retenus.

Tableau 2.1 : Caractéristiques principales des traitements retenus.

Traitement	Buts	Complexité	Nombre de paramètres	Liste des paramètres
Augmentation du contraste	<ul style="list-style-type: none"> <li>- accentuer les frontières entre les objets</li> <li>- éliminer des objets selon leur intensité</li> </ul>	$O(N)$	2	<ul style="list-style-type: none"> <li>- seuils minimal et maximal (nombres décimaux entre 0 et 1)</li> </ul>
Filtre de moyenne pondérée	<ul style="list-style-type: none"> <li>- lisser (bruit gaussien)</li> </ul>	$O(N K^3)$	5	<ul style="list-style-type: none"> <li>- poids des voisins immédiats en i, en j, en k, l et des autres voisins (nombre entiers positifs)</li> </ul>
Filtre médian	<ul style="list-style-type: none"> <li>- lisser (bruit impulsif)</li> </ul>	$O(N K^3 \log K^3)$	1	<ul style="list-style-type: none"> <li>- taille de la fenêtre du filtre (3 ou 5)</li> </ul>
Opération morphologique	<ul style="list-style-type: none"> <li>- modifier la taille des objets</li> <li>- fusionner / séparer</li> <li>- éliminer petits objets (clairs ou foncés)</li> </ul>	$O(N K^3)$	2	<ul style="list-style-type: none"> <li>- type d'opération (dilatation ou érosion)</li> <li>- type d'élément structurant (éléments 2x2x2 ou 3x3x3 pleins ou élément 3x3x3 en forme de croix)</li> </ul>
Croissance de région	<ul style="list-style-type: none"> <li>- segmenter</li> <li>- régulariser</li> </ul>	bornes $O(N)$ et $O(N K^3)$	2	<ul style="list-style-type: none"> <li>- position du germe de départ</li> <li>- seuils minimal et maximal (nombres décimaux entre 0 et 1)</li> </ul>

Les traitements proposés dans ce chapitre prennent leur intérêt, pour segmenter des images médicales, dans leur application séquentielle et ordonnée, qui sera présentée au chapitre des résultats. Le second objectif du projet, qui visait l'implantation sous la forme de filtres de traitements unitaires applicables sur des bains de voxels, a été atteint. La simplicité des traitements nous permet d'envisager la possibilité d'une implantation sur un processeur graphique (GPU), rapide et abordable. L'insertion dans un cadre logiciel de l'implantation (sur un processeur standard) des traitements sélectionnés est discutée au chapitre suivant portant sur la mise en oeuvre logicielle.

## CHAPITRE 3 : MISE EN OEUVRE LOGICIELLE

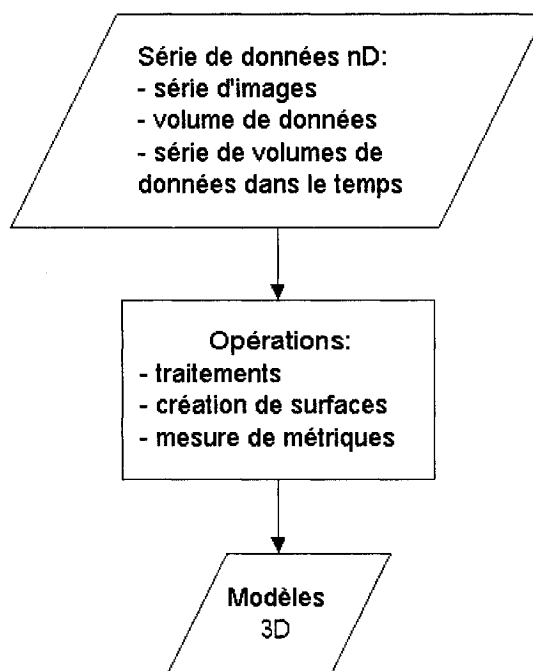
Ce chapitre expose la mise en oeuvre logicielle du projet : après avoir abordé les divers traitements retenus pour l'amélioration et la segmentation des données sources, nous allons poser un regard sur leur insertion dans le cadre global du logiciel développé. La première partie touche les aspects logiques de la conception du logiciel (philosophie générale, organisation des classes, choix des bibliothèques), c'est-à-dire l'architecture, alors que la seconde porte sur les aspects comportementaux, c'est-à-dire le fonctionnement (exécution) du logiciel.

### 3.1 Architecture logicielle

Tel que mentionné à la section 1.6, l'un des objectifs ayant trait à la conception logicielle du projet met l'accent sur : 1) le développement d'un cadre de travail facilitant l'ajout de nouvelles fonctionnalités ou méthodes alternatives, et 2) la favorisation de la réutilisation de design et de code pour produire un outil employable dans d'autres contextes d'application. Dans cette optique, nous avons conçu le logiciel dans un esprit de gabarit (*framework*). Un tel cadre de travail logiciel consiste, dans le paradigme orienté objet, en un ensemble de classes qui coopèrent afin de créer un modèle réutilisable pour une catégorie particulière de logiciels. Le gabarit est adapté à une application particulière en créant des classes spécifiques à l'application dérivées de ses classes abstraites, ces dernières définissant la structure globale et les principales responsabilités (Gamma, Helm, Johnson et Vlissides 1995). Bien que nous n'ayons pas conçu un réel gabarit suivant cette définition, ce qui aurait impliqué la génération automatique d'applications selon le contexte, nous avons retenu l'idée de généralisation et d'extensibilité pour la conception de notre application. Le système est une application standard qui nécessite la recompilation du code après l'ajout de nouvelles fonctionnalités.

Nous n'avons pas préconisé un style particulier d'architecture logicielle connue (ex. modèle client-serveur).

La catégorie particulière visée par notre projet comprend les logiciels de modélisation géométrique via des traitements appliqués sur des séries de données suivant le schéma présenté à la figure 3.1.



**Figure 3.1 : Flux global d'opérations de la catégorie particulière de logiciels visée.**

Le logiciel reçoit en entrée une série de données qui peut être en 3D ou en 4D. Dans notre application, la modélisation de vaisseaux sanguins à partir d'images médicales se fait en trois dimensions, mais le logiciel a été conçu en fonction de la possibilité de traiter des données en quatre dimensions. Il serait en effet fort pertinent d'inclure la dimension temporelle afin de retrouver plusieurs modèles à différents temps. Un autre exemple de volume de données pourrait être des données de simulation numérique pour la modélisation de nuages 3D de concentrations de gaz dans une pièce, ce genre d'application faisant partie de la catégorie de logiciels visée. Nous avons défini un format de fichier pour les données sources, le format « VOL », pouvant tenir compte de

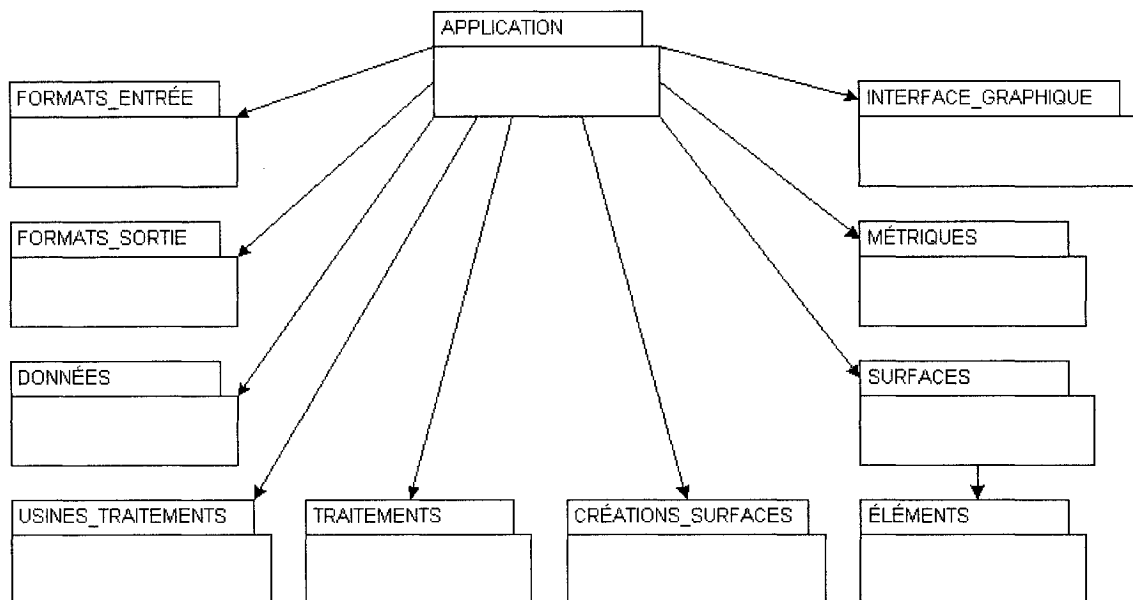


cette liberté au niveau du nombre de dimensions. Sa structure est présentée à l'annexe C. Les images médicales utilisées, produites par IRM, sont stockées sous le standard DICOM (National Electrical Manufacturers Association 2006) et nécessitent donc une conversion vers le format « VOL », effectuée conjointement à l'aide d'un court programme que nous avons développé et du logiciel ezDICOM (Rorden 2002).

Le coeur du programme consiste à faire des opérations qui se divisent en trois catégories : 1) traitements sur les données sources; 2) création de surfaces 3D à partir des données traitées; 3) mesures de métriques sur les surfaces créées. La plus importante catégorie est sans doute la première, qui a fait l'objet du chapitre précédent. Le logiciel a été développé pour pouvoir être utilisé en mode interactif, avec une interface graphique, de même qu'en mode console, l'utilisateur fournissant dans ce cas la liste des traitements à effectuer dans un fichier. Nous avons créé le format de fichier « TRM » à cette fin, dont la structure est donnée à l'annexe C.

Le logiciel exporte en sortie des surfaces tridimensionnelles modélisant l'organe sous forme de maillages de polygones. Nous avons choisi d'implanter le support de deux types de fichiers de sortie, le format « VU » et le format « YAMS », dont les structures sont présentées à l'annexe C. VU est un logiciel de visualisation scientifique (inVisu 2005) et YAMS un logiciel de remaillage adaptatif (Institut National de Recherche en Informatique et en Automatique 2004).

La figure 3.2 illustre l'organisation à haut niveau des sous-systèmes du programme. Le logiciel, développé en C++, comporte une cinquantaine de classes que l'on peut regrouper en 11 sous-systèmes, et comprend environ 11 000 lignes de code source. Chaque sous-système comprend plusieurs classes ayant des responsabilités et fonctionnalités semblables et parfois situées de la même hiérarchie, c'est-à-dire liées par des relations de généralisation. Les flèches de la figure représentent les relations les plus importantes entre les sous-systèmes. Ces relations sont de type agrégation et association.



**Figure 3.2 : Organisation des sous-systèmes du programme.**

Plusieurs des sous-systèmes comprennent une hiérarchie de classes, avec une classe abstraite en tête. Pour ajouter une nouvelle technique (ex. un nouveau traitement à appliquer sur les données) ou un nouveau format supporté (ex. un nouveau format de sortie), il suffit ainsi de créer une nouvelle classe dérivée dans la structure hiérarchique, similaire aux autres classes dérivées existantes. Les interactions entre les sous-systèmes se font généralement à travers les classes abstraites qui fournissent l'interface de communication. L'impact d'ajouts de nouvelles techniques ou nouveaux formats sur les autres sous-systèmes s'en trouve également limité. Les sections suivantes introduisent les différents sous-systèmes. L'annexe D présente un diagramme illustrant la structure de chacun des sous-systèmes.

### 3.1.1 Sous-système « Application »

Le sous-système « Application » se situe au coeur du programme. Il permet l'interaction entre tous les autres sous-systèmes, il est responsable de coordonner l'exécution du

programme et de stocker un pointeur vers les divers objets utilisés (relations d'agrégation) : le format de fichier des données sources, l'usine permettant de créer les traitements pour le bon nombre de dimensions des données, les données à traiter ainsi qu'une copie des données originales, le traitement courant, la méthode de création de surface, la surface représentant le modèle, la métrique courante pour valider la surface, de même que le format de fichier de sortie permettant d'exporter la surface. Il connaît le nombre de dimensions des données et s'occupe de créer les bons types d'objets via l'usine.

La classe de base « Application » ne peut être utilisée en soi. Deux classes, dont il est possible de créer des instances, en sont dérivées : « ApplicationConsole » et « ApplicationGraphique ». La première est utilisée en mode console. Sa particularité réside dans le fait qu'elle récupère les traitements à effectuer sur les données à partir d'un fichier « TRM ». La seconde est utilisée en mode interactif. Elle est responsable de l'affichage des données et des surfaces et fait le lien entre le sous-système d'interface graphique et la structure de données.

### **3.1.2 Sous-système « Interface Graphique »**

Le sous-système « Interface Graphique » diffère des autres sous-systèmes du fait qu'il ne comporte pas de hiérarchie de classes. Absent du mode console, il gère l'affichage des composantes de l'interface usager graphique et les événements de la souris et du clavier. Il fait appel à la boîte à outils GLOW (Azuma 2000), une interface de programmation orientée objet basée sur GLUT (OpenGL 2004) fournissant les mécanismes de création de fenêtres graphiques, de menus et autres composantes graphiques et de gestion des événements pour les applications utilisant la bibliothèque graphique OpenGL. Malgré le fait que la boîte à outils GLOW ne soit plus en développement depuis quelques années, nous avons retenu ses services car elle offre l'intéressant avantage d'être fournie sous forme de code source (environ 29 000 lignes

de code source) qui peut être directement intégré à l'ensemble du code et modifié au besoin.

En plus des classes de GLOW, le sous-système contient la classe « FenetrePrincipale », qui hérite de plusieurs classes de GLOW afin d'implanter les diverses fonctionnalités d'une fenêtre principale d'une application graphique. Cette classe possède un pointeur vers l'application graphique, ce qui lui permet de déclencher les opérations requises à la suite d'événements de souris ou de clavier. Les sous-systèmes « Application » et « Interface Graphique » peuvent donc communiquer de façon bidirectionnelle.

### **3.1.3 Sous-systèmes « Formats Entrée » / « Formats Sortie »**

Les sous-systèmes « Formats Entrée » et « Formats Sortie » sont très semblables en structure et en responsabilité. Ils font la transition entre les fichiers de données et la structure de données du programme. Ils fournissent des interfaces communes à tous les formats d'entrée et à tous les formats de sortie à travers les méthodes des classes abstraites de base « FormatEntree » et « FormatSortie ». Chaque sous-système comprend une structure hiérarchisée de classes, les classes dérivées implantant la lecture et / ou l'écriture dans un format particulier. Ces formats sont le format « VOL » en entrée et les formats « VU » et « YAMS » en sortie. Cette conception facilite l'ajout de nouveaux formats de fichiers d'entrée ou de sortie. La structure de la hiérarchie des classes est inspirée du patron de conception « Stratégie » (voir le patron comportemental *Strategy* dans Gamma, Helm, Johnson et Vlissides 1995).

### **3.1.4 Sous-système « Données »**

Le sous-système « Données » est responsable de la structure de données. Il stocke les voxels, fournit un moyen de leur accéder sans révéler inutilement la structure interne et

s'occupe de leur affichage dans le mode interactif. Afin d'accommoder les données 3D et l'éventuelle possibilité de données 4D, le sous-système possède une hiérarchie de classes avec à sa tête la classe abstraite « Donnees », et deux classes dérivées, « Donnees3D » et « Donnees4D ». La classe abstraite conserve les dimensions réelles des voxels et un pointeur vers un tableau unidimensionnel contenant les valeurs des voxels. Les classes dérivées remplissent ce tableau tout en sachant comment les différentes dimensions sont organisées. Il aurait été possible de n'avoir qu'une seule classe qui supporte le nombre maximal de dimensions, mais puisque des données 3D doivent s'afficher différemment des données 4D (le passage à la quatrième dimension n'est pas automatique), nous avons jugé bon de les séparer.

### **3.1.5 Sous-systèmes « Usines Traitements » / « Traitements »**

Le sous-système « Traitements », l'un des sous-systèmes clés du logiciel, s'occupe d'effectuer les traitements proposés au chapitre 2 sur les données sources. Il comporte une hiérarchie de classes à trois niveaux, inspirée du patron de conception « Stratégie » : au premier niveau se trouve la classe abstraite « Traitement », qui fournit une interface d'utilisation au client; au second niveau se trouvent plusieurs classes abstraites dérivées, une pour chaque type de traitement (augmentation de contraste, filtre de moyenne pondérée, etc.); enfin, au troisième niveau, deux classes concrètes sont dérivées de chaque classe du second niveau afin de permettre une implantation différente pour des opérations sur des données 3D et 4D. En effet, l'ajout de la quatrième dimension augmente souvent la complexité d'un algorithme de façon dramatique, nécessitant le recours à des mécanismes de calculs alternatifs. Chaque classe abstraite spécialisée pour un traitement en particulier conserve les paramètres utilisés, reçus à la construction par l'une des deux classes concrètes dérivées. À noter que l'interface des classes en 4D a été conçue mais seules les classes en 3D ont été implantées.

Afin de créer la bonne famille de traitements (3D ou 4D), le patron de conception « Usine Abstraite » (voir le patron de création *Abstract Factory* dans Gamma, Helm, Johnson et Vlissides 1995) a été retenu. Le sous-système « Usines Traitements » sert à configurer le système avec une seule des deux familles de traitements. Il est organisé en hiérarchie de classes, avec à sa tête la classe « UsineAbsTraitements », qui déclare une interface pour les opérations de création des produits abstraits (classes « TraitementA », « TraitementB », etc. du second niveau de la hiérarchie du sous-système « Traitement »), suivant le patron de conception « Usine » (voir le patron de création *Factory Method* dans Gamma, Helm, Johnson et Vlissides 1995). Deux classes concrètes en héritent, « UsineTraitements3D » et « UsineTraitements4D », qui implantent les opérations de création des produits concrets (« TraitementA3D », « TraitementA4D », etc. du troisième niveau de la hiérarchie).

### **3.1.6 Sous-système « Créations Surfaces »**

Le sous-système « Créations Surfaces » suit la même logique structurelle que les sous-systèmes « Formats Entrée » et « Formats Sortie ». Responsable de la création de la surface polygonale à partir des voxels traités, il est organisé en hiérarchie de classes avec une classe de base abstraite fournissant l'interface de communication et une classe dérivée concrète qui implante une technique particulière de création de surface, à la manière du patron de conception « Stratégie ». Une seule technique de création de surface a été implantée pour les besoins du projet : les cubes baladeurs, qui créent une surface 3D triangulée (voir la section 1.3.2). Cette conception facilite l'ajout de nouvelles techniques de création de surfaces.

### 3.1.7 Sous-systèmes « Surfaces » / « Éléments »

Le sous-système « Surfaces » sert à stocker des surfaces et les afficher en mode interactif. Il est conçu de manière hiérarchique, à l'instar du sous-système « Données », avec une classe de base abstraite « Surface » fournissant les mécanismes d'accès à sa structure de données : vecteurs d'éléments polygonaux, d'arêtes, de sommets et de normales, implantés avec STL (voir Silicon Graphics, Inc. 2005). L'abstraction vient du fait que la surface générale ne sait pas comment elle doit s'afficher, ce qui est défini dans la classe dérivée concrète, une surface tridimensionnelle triangulée dans le cas de ce projet. Le sous-système « Éléments » va de pair avec le sous-système « Surfaces » car, organisé également de façon hiérarchique, il permet à la classe « Surface » de conserver un vecteur de pointeurs vers des éléments abstraits, qui se concrétisent dans ce cas en triangles. Cette conception offre beaucoup de latitude quant au type de surfaces.

### 3.1.8 Sous-système « Métriques »

Le sous-système « Métriques » est responsable de calculer diverses mesures sur une surface dans un but de validation. Il s'organise de façon hiérarchique selon le patron « Stratégie » à la manière des sous-systèmes « Formats Entrée », « Formats Sortie » et « Créations Surfaces » pour faciliter l'ajout de nouvelles métriques. La classe de base abstraite « Metrique » procure l'interface de calcul pour le client, et le calcul de différentes métriques est encapsulé dans des classes séparées, héritant de la classe abstraite. Nous avons choisi d'implanter le calcul de l'aire de la surface, c'est-à-dire la somme de l'aire des éléments polygonaux de la surface et le calcul du volume intérieur de la surface. Ces calculs sont inspirés du code source du *Visualization Toolkit*, un outil logiciel pour l'infographie 3D, le traitement d'images et la visualisation (Kitware 2006).

### 3.2 Aspects comportementaux

Le programme développé pour l'application particulière visée, la modélisation 3D de vaisseaux sanguins à partir d'images médicales, a été baptisé « Architecte de Segmentation de Séries d'Images Médicales et de Modélisation d'Organes 3D » (ASSIMMO3D). Les points suivants regroupent les caractéristiques importantes du logiciel :

- Il doit pouvoir fonctionner en deux modes distincts :
  - console, pour lequel tous les paramètres sont spécifiés par l'utilisateur au départ et les traitements à effectuer sont écrits dans un fichier;
  - interactif, avec une interface usager graphique.
- Il doit lire les données d'un fichier d'entrée.
- Il doit permettre l'application de traitements sur les données.
- Les traitements disponibles doivent pouvoir être appliqués individuellement, dans n'importe quel ordre et pour un nombre quelconque de fois.
- Le logiciel doit être capable de créer une isosurface à partir des données (traitées ou non).
- Il doit permettre de mesurer quelques métriques (aire et volume interne) pour la validation de la surface.
- La surface produite doit pouvoir être sauvegardée dans un fichier de sortie.

Les points suivants regroupent les caractéristiques importantes du logiciel en mode interactif :

- L'utilisateur doit pouvoir sélectionner le fichier d'entrée.
- Les données sont affichées à l'écran de manière à permettre une visualisation selon différents plans en différentes positions.
- L'utilisateur doit pouvoir choisir de faire afficher les données modifiées (en cours de traitement) ou les données sources.
- Il doit être possible de sauvegarder les données modifiées dans le format d'entrée.



- La sélection des paramètres pour les traitements se fait à l'aide de boîtes de dialogue.
- Une fois que le fichier d'entrée a été ouvert, l'utilisateur peut créer la surface à tout moment. Il peut continuer d'appliquer des traitements sur les données même si la surface a déjà été créée.
- Lorsqu'une surface a été créée, elle peut être affichée en superposition à un plan des données sources.
- L'utilisateur doit pouvoir sélectionner le fichier de sortie.

Les deux modes d'exécution du logiciel s'utilisent dans des contextes différents. Le mode interactif est à favoriser lorsque l'utilisateur désire explorer et visualiser les données et les effets de divers traitements, alors que le mode console permet d'effectuer des opérations en série, sans nécessiter la présence de l'utilisateur durant l'exécution.

Le mode d'utilisation est déterminé au lancement de l'exécutable du programme : s'il est lancé avec des arguments, la version console sera démarrée, alors que si aucun argument n'est passé au programme, la version interactive sera démarrée. Les paramètres à fournir en mode console concernent le nom du fichier d'entrée (il n'y a pour l'instant qu'un seul type de format, « VOL »), le nom du fichier de commandes « TRM », les coordonnées d'un voxel situé à l'intérieur de l'objet à modéliser, le format de sortie et le nom du fichier de sortie. Les détails de leur utilisation sont présentés à la figure E.1 de l'annexe E. Afin d'illustrer les interactions entre les divers objets, un diagramme de séquence UML pour le mode console est également donné à la figure E.2 de l'annexe E. Certains objets ont été laissés de côté (les métriques, etc.) afin d'alléger le diagramme.

En mode interactif, l'utilisateur fera par exemple les manipulations suivantes. Une fois le programme lancé, il doit créer un nouveau projet ou ouvrir un fichier « VOL » dont les valeurs des voxels ont déjà subi des traitements. Lorsqu'un projet est ouvert, les données sur les voxels sont chargées en mémoire et, dans le cas 3D, le premier plan image

(perpendiculaire à l'axe des z) est affiché à l'intérieur d'une boîte délimitant le volume de données, dans la fenêtre principale. Une seconde fenêtre apparaît avec plusieurs contrôles sur l'affichage : la position de la caméra, le niveau de zoom, le choix du plan d'affichage et de sa position, l'animation des plans d'affichage, etc. (voir la figure E.3 de l'annexe E pour une image de cette fenêtre). L'utilisateur peut visualiser les données dans l'un des trois plans (i-j, i-k, j-k), une tranche à la fois, à la position désirée. La figure 3.3 présente des exemples de visualisation d'images médicales selon trois coupes différentes : axiale, coronale et sagittale.

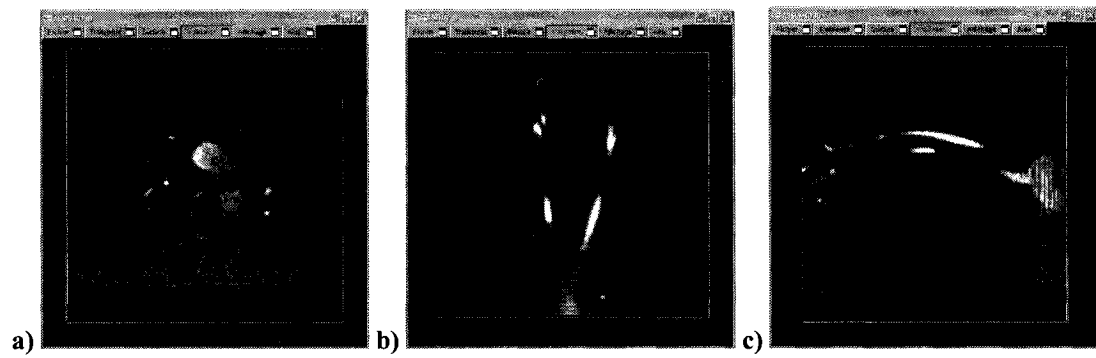


Figure 3.3 : Visualisation d'images médicales selon trois coupes : a) axiale, b) coronale et c) sagittale.

L'utilisateur, à travers le menu « Traitement », peut choisir le traitement à appliquer sur les données parmi ceux présentés au chapitre 2. Lorsqu'un traitement est sélectionné, une nouvelle fenêtre (boîte de dialogue) apparaît temporairement afin de permettre à l'utilisateur de choisir les paramètres spécifiques au traitement. L'interface usager graphique de ces boîtes de dialogue est donnée à la figure E.4 de l'annexe E. À tout moment, l'utilisateur peut choisir de créer une surface par la méthode des cubes baladeurs, avec une isovaleur par défaut à 0,5. S'il n'est pas satisfait de la surface, d'autres traitements peuvent être appliqués sur les données avant de créer une nouvelle surface. Il est possible d'afficher les données traitées de même que les données originales, ce qui permet des comparaisons visuelles. L'utilisateur peut choisir d'enregistrer les données traitées ou partiellement traitées sous le format « VOL » afin de poursuivre les traitements pendant une autre séance. Lorsqu'une surface a été créée, le menu « Validation » devient

disponible, offrant les calculs de l'aire et du volume de la surface. L'utilisateur peut alors exporter cette surface dans le format « VU » ou le format « YAMS » et l'utiliser comme il le souhaite.

### **3.3 Autres considérations**

La phase de test et validation du logiciel, c'est-à-dire la vérification des aspects de logique interne et de fonctionnalité externe, n'a pas été menée de façon exhaustive. Pour tous les résultats présentés au chapitre suivant de même que pour d'autres cas qui n'ont pas été retenus dans ce mémoire, le système a montré un comportement en accord avec ce qui était attendu. La performance de l'application et la complexité des opérations sont abordées à la section 4.4 et discutées à la section 5.4.

La facilité attendue quant à l'ajout de nouvelles méthodes a été constatée lors du développement du logiciel. L'application était fonctionnelle après l'implantation des deux premiers traitements (augmentation du contraste et filtre de moyenne pondérée). Les autres traitements ont été ajoutés progressivement et aisément en suivant le modèle des classes existantes pour mener à l'application complète à laquelle ce mémoire se consacre.

En ce qui a trait à l'utilisation de la mémoire, la partie variable d'une exécution à l'autre concerne le chargement des données sources. Ces données sont conservées dans un format à virgule flottante de simple précision, ce qui occupe, dans le cas le plus volumineux de 140 images médicales d'une taille de 256 x 256 voxels, environ 36 Mo.

Le traitement des erreurs possibles à l'exécution du programme est présent mais minimal. Par exemple, les opérations de division par zéro possibles sont testées. Les arguments passés au programme en mode console et ainsi que les valeurs des paramètres des traitements sont vérifiés et l'application prend une décision en cas d'erreur en

mettant fin à l'exécution ou en forçant une valeur par défaut, tout en affichant le message d'erreur.

L'application ne dépend pas d'un système d'exploitation en particulier. Elle a été développée sous Windows XP mais pourrait être compilée et exécutée sur Linux. Le fait que la librairie GLOW soit fournie sous forme de code source contribue à la portabilité de l'application.

Enfin, les commentaires situés aux endroits clés dans le code, de même que les explications présentes dans les entêtes des classes et des fonctions, constituent l'essentiel de la documentation sur l'application.

-----

Dans ce chapitre, nous avons présenté l'architecture logicielle du projet et son fonctionnement. Codé en C++, le logiciel développé peut être utilisé en deux modes distincts : console et interactif. L'encapsulation de la majorité des fonctionnalités dans des classes séparées, membres de hiérarchies (ex. traitements, formats de fichiers supportés, création de surfaces), et l'interaction entre sous-systèmes via les classes de base des hiérarchies offre la possibilité d'ajouter facilement de nouvelles méthodes et fonctionnalité. Le premier objectif, posé au chapitre 1, qui visait à développer un cadre de travail permettant de tester plusieurs algorithmes et facilitant l'ajout de nouvelles fonctionnalités ou méthodes alternatives, et à favoriser la réutilisation de design et de code pour produire un outil employable dans d'autres contextes d'application, a été atteint.

Nous allons voir des exemples concrets d'utilisation du logiciel conçu et introduit dans ce chapitre à travers les résultats présentés au chapitre suivant.

## CHAPITRE 4 : RÉSULTATS

Ce chapitre présente les résultats du projet à travers l'application spécifique de modélisation 3D de vaisseaux sanguins à partir d'images médicales. Trois séries de données ont été retenues pour illustrer l'utilisation de séquences de filtres 3D :

1. images médicales de la crosse aortique et des carotides,
2. images médicales de l'aorte,
3. données analytiques représentant la crosse aortique et l'aorte.

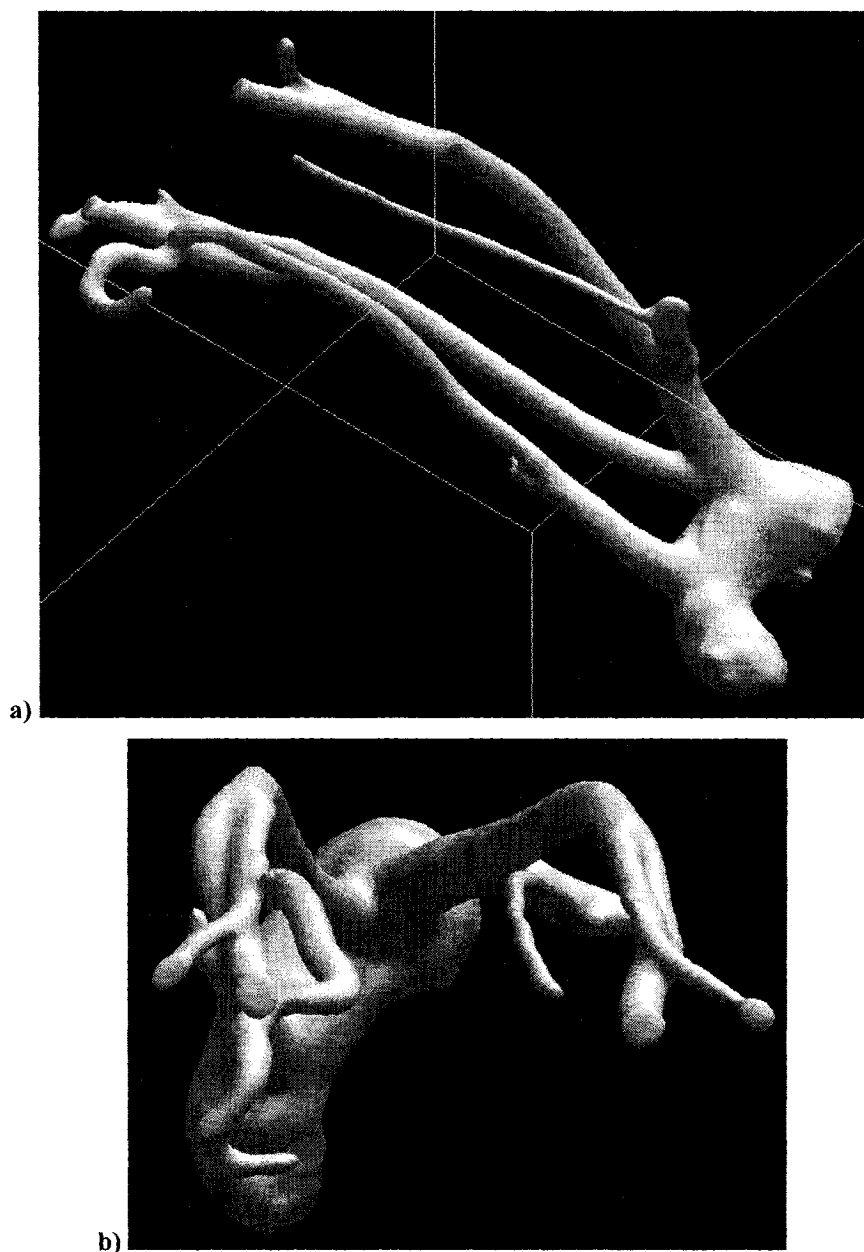
Les prochaines sections fournissent, pour chaque série, une brève description des données, les images des surfaces résultantes telles que produites par notre logiciel, la séquence de traitements ayant servi à la modélisation des surfaces de même que leurs paramètres, ainsi que les mesures d'aire et de volume interne des surfaces. Nous discuterons de la composition des séquences de traitements au chapitre suivant. Les temps d'exécution des traitements pour les différentes séries sont par la suite regroupés à la section suivante, puis enfin quelques images de l'exportation des surfaces en format « VU » et « YAMS » terminent ce chapitre.

### 4.1 Série crosse aortique et carotides

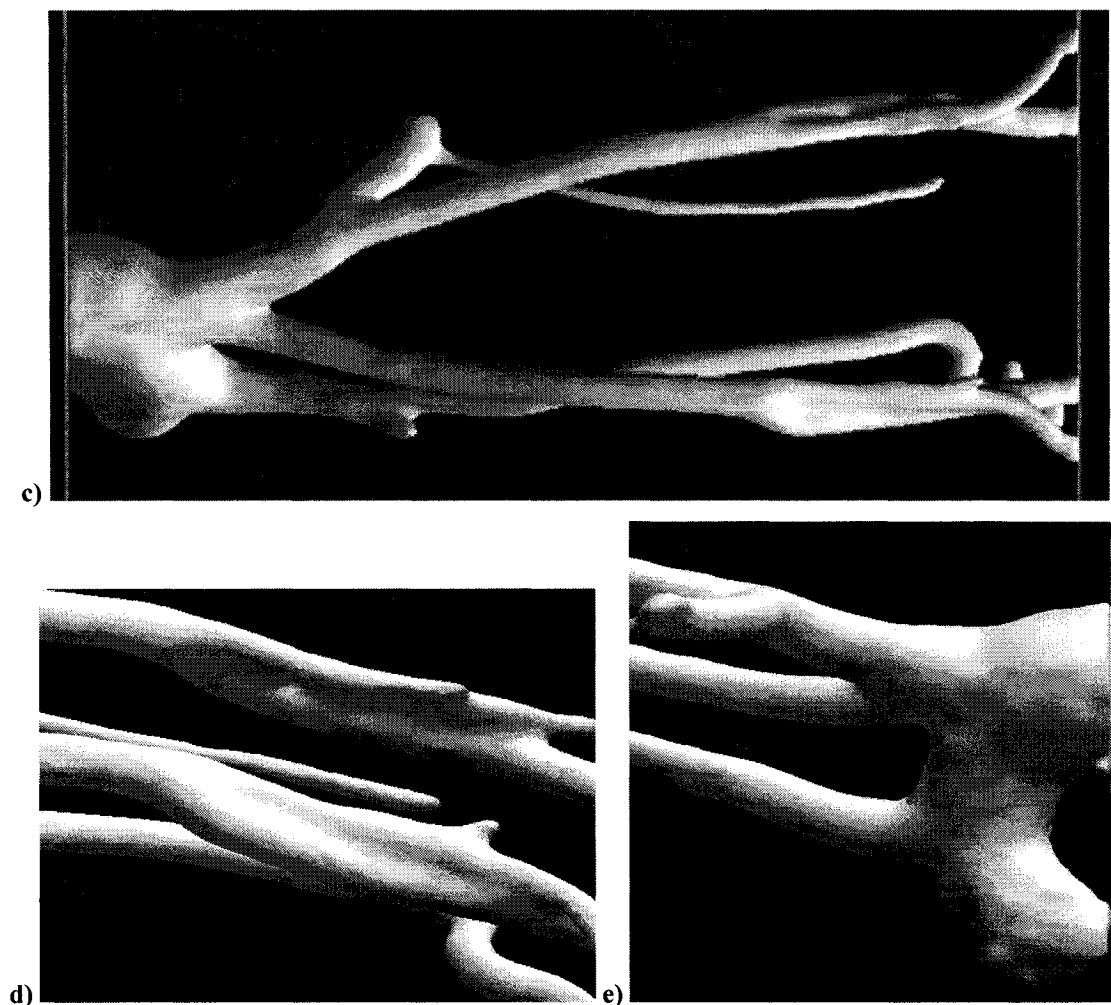
Cette série de données a été produite via imagerie par résonance magnétique selon la technique d'angiographie. Elle est constituée de coupes axiales (horizontales par rapport à un individu en position debout) débutant au niveau du thorax, légèrement au-dessus du cœur, et se terminant au niveau de la tête, légèrement au-dessus du menton. La série comporte 140 images médicales, produites avec une résolution de 256 x 256 pixels d'une taille physique d'un peu moins d'un millimètre. Les coupes sont espacées de 1,5 millimètres, ce qui constitue un ensemble anisotrope de voxels. La série complète est montrée à l'annexe A. Cette série regroupe plusieurs difficultés majeures : la détection

de la crosse aortique, dont les voxels présentent des intensités très variables, et des nombreuses bifurcations des carotides.

Les figures 4.1a à 4.1e montrent les surfaces obtenues pour cette série de données, produites après l'application de la séquence de traitements présentée à la figure 4.2.



**Figure 4.1 : Surfaces 3D pour la série crosse aortique et carotides. En a), vue d'ensemble de biais; en b), vue d'ensemble axiale.**



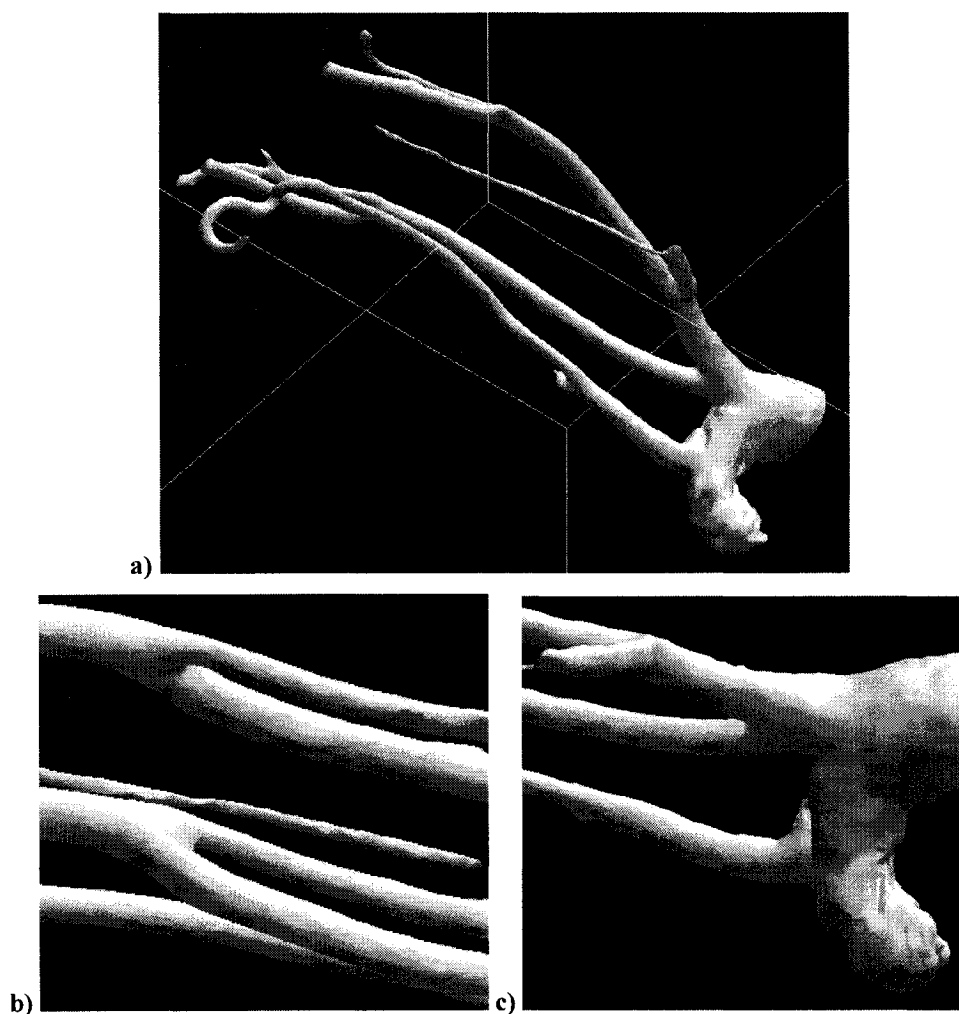
**Figure 4.1 (suite) : En c), vue d'ensemble coronale; en d) vue sagittale des carotides; en e) vue sagittale de la crosse aortique.**

- 1) Augmentation du contraste (seuils minimal et maximal : 0,3 et 0,5)
- 2) Filtre médian (noyau de taille 5 x 5 x 5)
- 3) Fermeture morphologique (élément structurant cubique de 3 x 3 x 3)
- 4) Filtre de moyenne pondérée (poids des voisins immédiats en i, en j et en k et poids des autres voisins : 1)
- 5) Croissance de région (germe aux coordonnées 125, 160, 0; seuils minimal et maximal : 0,2 et 1,0)
- 6) Augmentation du contraste (seuils minimal et maximal : 0,1 et 0,3)
- 7) Filtre de moyenne pondérée (poids des voisins immédiats en i, en j et en k et poids des autres voisins : 1)

**Figure 4.2 : Séquence de traitements pour la série crosse aortique et carotides.**

La surface a été produite avec une isovaleur de 0,5, la plage de valeurs possibles allant de 0,0 à 1,0. Deux plans images aux intensités nulles ont été ajoutés au début et à la fin de la série d'images médicales afin de produire des surfaces fermées.

D'autres tests sont également présentés car les séquences de traitements et les résultats peuvent varier substantiellement selon l'objectif recherché. Les figures 4.3a à 4.3c montrent les surfaces obtenues pour la même série de données, produites après l'application de la séquence de traitements présentée à la figure 4.4, favorisant la délimitation du bout des carotides.



**Figure 4.3 : Surfaces 3D pour la série crosse aortique et carotides, cas # 2. En a) vue d'ensemble de biais; en b), vue sagittale des carotides; en c), vue sagittale de la crosse aortique.**

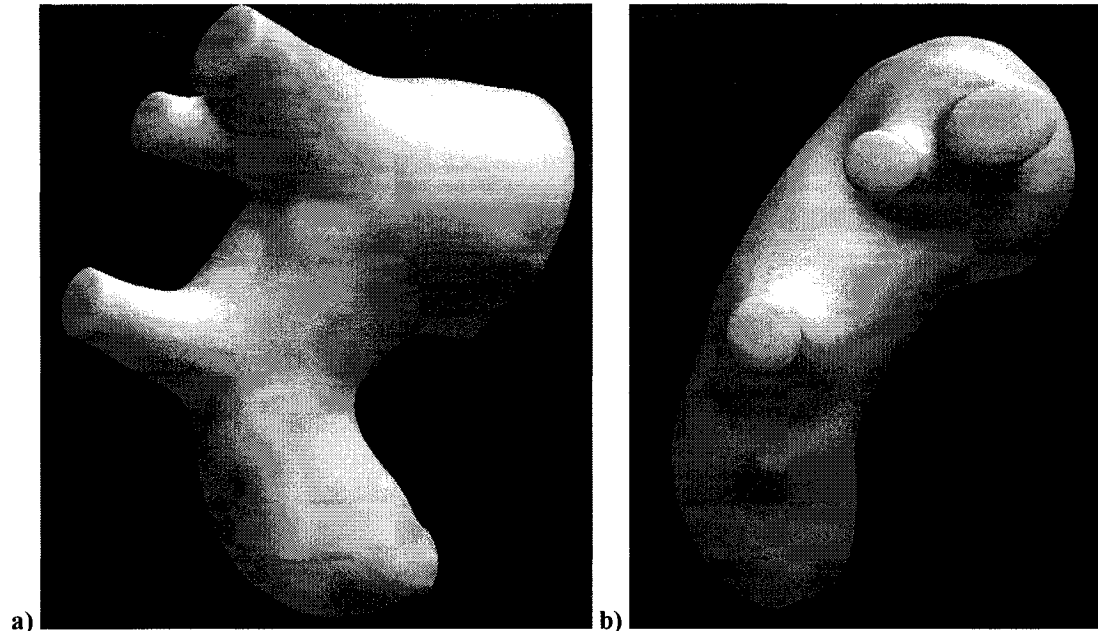


- 1) Augmentation du contraste (seuils minimal et maximal : 0,4 et 0,6)
- 2) Croissance de région (germe aux coordonnées 125, 160, 0;  
seuils minimal et maximal : 0,2 et 1,0)
- 3) Filtre de moyenne pondérée  
(poids des voisins immédiats en i, en j et en k : 1;  
poids des autres voisins : 0)

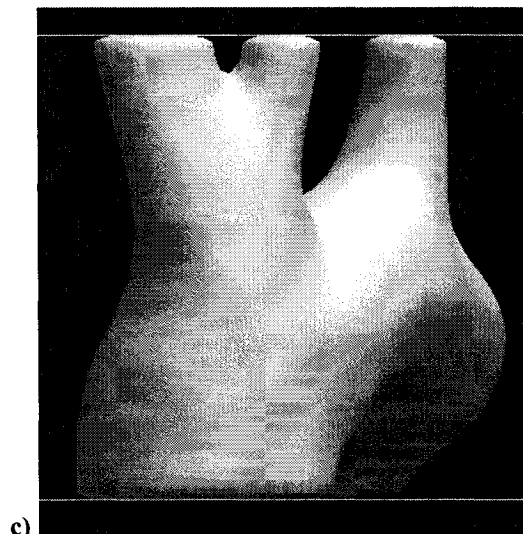
**Figure 4.4 : Séquence de traitements pour la série crosse aortique et carotides, cas # 2.**

La surface a été produite avec une isovaleur de 0,5. Deux plans images aux intensités nulles ont été ajoutés au début et à la fin de la série d'images médicales afin de produire des surfaces fermées.

Les figures 4.5a à 4.5c montrent les surfaces obtenues pour la même série de données mais tronquée à la 30<sup>e</sup> image. Les surfaces ont été produites après l'application de la séquence de traitements présentée à la figure 4.6, favorisant la délimitation de la crosse aortique.



**Figure 4.5 : Surfaces 3D pour la série crosse aortique et carotides tronquée, cas # 3. En a), vue de biais de la crosse aortique; en b), vue axiale de la crosse aortique.**



c)

**Figure 4.5 (suite) : En c), vue coronale de la crosse aortique.**

- 1) Filtre médian (noyau de taille 5 x 5 x 5)
- 2) Filtre médian (noyau de taille 5 x 5 x 5)
- 3) Fermeture morphologique (élément structurant cubique de 3 x 3 x 3)
- 4) Filtre de moyenne pondérée  
(poids des voisins immédiats en i et en j : 2;  
poids des voisins immédiats en k : 3;  
poids des autres voisins : 1)
- 5) Croissance de région (germe aux coordonnées 125, 160, 0;  
seuils minimal et maximal : 0,3 et 1,0)
- 6) Augmentation du contraste (seuils minimal et maximal : 0,2 et 0,4)
- 7) Filtre de moyenne pondérée  
(poids des voisins immédiats en i, en j et en k : 2;  
poids des autres voisins : 1)
- 8) Filtre de moyenne pondérée  
(poids des voisins immédiats en i, en j et en k : 2;  
poids des autres voisins : 1)
- 9) Filtre de moyenne pondérée  
(poids des voisins immédiats en i, en j et en k : 2;  
poids des autres voisins : 1)

**Figure 4.6 : Séquence de traitements pour la série crosse aortique et carotides, cas # 3.**

La surface a été produite avec une isovaleur de 0,5. Deux plans images aux intensités nulles ont été ajoutés au début et à la fin de la série d'images médicales afin de produire des surfaces fermées.

Le tableau 4.1 donne les informations relatives aux surfaces obtenues pour les trois cas présentés.

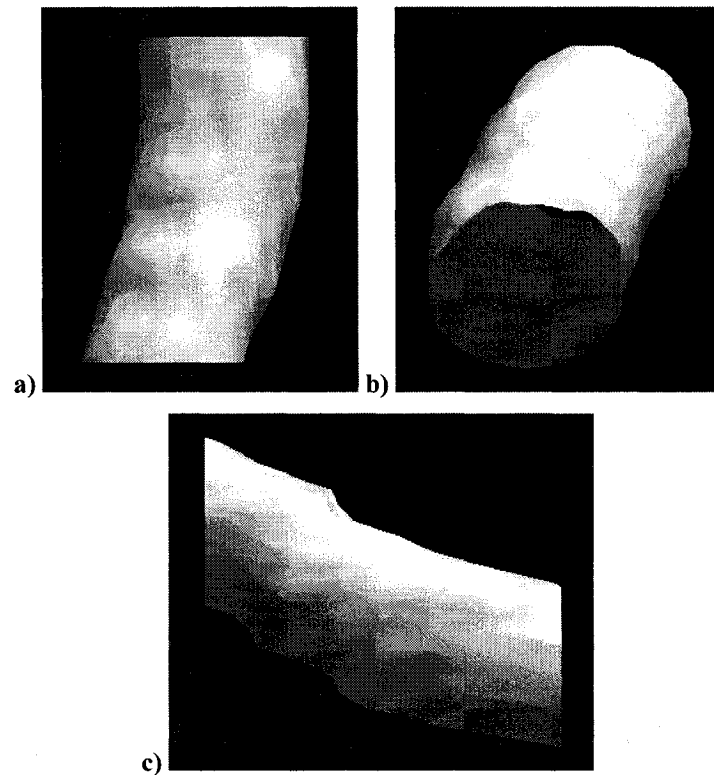
**Tableau 4.1 : Informations relatives à la surface pour la série crosse aortique et carotides.**

	<b>Cas #1 (but général)</b>	<b>Cas #2 (attention sur les carotides)</b>	<b>Cas #3 (attention sur la crosse aortique, 30 des 140 images)</b>
<b>Nombre de sommets</b>	27 731	23 129	8 552
<b>Nombre d'éléments (triangles)</b>	55 454	46 278	17 100
<b>Aire de la surface (mm<sup>2</sup>)</b>	21 747	17 940	6 394
<b>Volume interne de la surface (mm<sup>3</sup>)</b>	55 350	35 911	27 995

## 4.2 Série aorte

À l'instar de la série crosse aortique et carotides, cette série de données a été produite via imagerie par résonance magnétique. Elle est utilisée dans le but d'obtenir une idée générale de l'allure des organes. Le système vasculaire n'est pas tellement bien contrasté par rapport aux autres organes voisins. Cette série est constituée de coupes axiales (horizontales par rapport à un individu en position debout) situées au niveau du thorax. La série comporte seulement 5 images médicales, produites avec une résolution de 256 x 256 pixels d'une taille physique d'un peu moins de deux millimètres. Les coupes sont espacées de 10 millimètres, ce qui constitue un ensemble anisotrope de voxels beaucoup plus prononcé que pour la série précédente. La série complète est montrée à l'annexe A. Cette série a été retenue car elle regroupe un autre type de difficultés : malgré que l'allure de l'aorte soit relativement simple (de forme cylindrique) et bien définie, les variations substantielles de sa position dans le plan image d'une image à l'autre peuvent engendrer la récupération d'objets non désirés. En effet, les voxels appartenant à des objets d'intensité semblable à celle de l'aorte peuvent se retrouver immédiatement voisins d'un voxel appartenant à l'aorte.

Les figures 4.7a à 4.7c montrent les surfaces obtenues pour cette série de données, produites après l'application de la séquence de traitements présentée à la figure 4.8.



**Figure 4.7 : Surfaces 3D pour la série aorte. En a), vue coronale; en b), vue axiale; en c), vue sagittale.**

- 1) Augmentation du contraste (seuils minimal et maximal : 0,6 et 0,8)
- 2) Croissance de région (germe aux coordonnées 115, 110, 0; seuils minimal et maximal : 0,9 et 1,0)
- 3) Filtre de moyenne pondérée  
(poids des voisins immédiats en i et en j : 1;  
poids des voisins immédiats en k et poids des autres voisins : 0)
- 4) Filtre de moyenne pondérée  
(poids des voisins immédiats en i et en j : 1;  
poids des voisins immédiats en k et poids des autres voisins : 0)
- 5) Augmentation du contraste (seuils minimal et maximal : 0,1 et 0,3)
- 6) Filtre de moyenne pondérée  
(poids des voisins immédiats en i et en j : 1;  
poids des voisins immédiats en k et poids des autres voisins : 0)

**Figure 4.8 : Séquence de traitements pour la série aorte.**

La surface a été produite avec une isovaleur de 0,5, la plage de valeurs possibles allant de 0,0 à 1,0. Des plans images aux intensités nulles n'ont pas été ajoutés au début et à la fin de la série d'images médicales comme pour la série crosse aortique et carotides car vu le petit nombre d'images sources, l'impact des plans ajoutés pour fermer la surface aurait été trop grand. Le tableau 4.2 donne les informations relatives à la surface obtenue. Le volume interne n'est pas calculé car la surface est ouverte.

**Tableau 4.2 : Informations relatives à la surface pour la série aorte.**

<b>Nombre de sommets</b>	480
<b>Nombre d'éléments (triangles)</b>	872
<b>Aire de la surface (mm<sup>2</sup>)</b>	2 813

### 4.3 Données analytiques

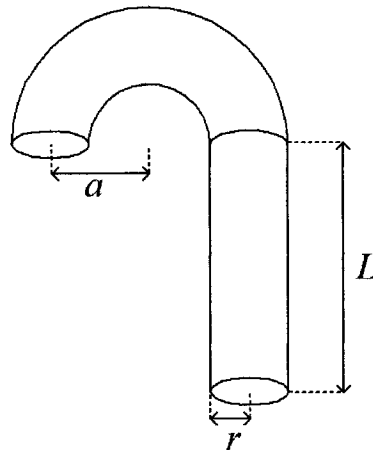
Cette série de données est issue d'un processus de discrétisation, sous forme de voxels, d'une surface définie mathématiquement, représentant la crosse aortique et l'aorte (Lincourt-Éthier 2005). Elle est formée d'un demi-tore et d'un cylindre de même diamètre, positionnés dans le centre d'un bain de voxels de dimensions 128 x 128 x 128. La taille des voxels est isotrope, d'un peu moins d'un dixième d'unité pour chaque direction. Chaque voxel peut avoir l'une des deux valeurs possibles : 0,0 s'il se trouve à l'extérieur de l'objet ou 1,0 s'il se trouve sur la frontière ou à l'intérieur de l'objet. L'aire de la surface et le volume interne sont connus. Ils sont calculés de la manière suivante :

$$Aire = (2\pi rL) + (2\pi^2 ra) + 2(\pi r^2) \quad (4.1)$$

$$Volume = (\pi r^2 L) + (\pi^2 r^2 a) \quad (4.2)$$

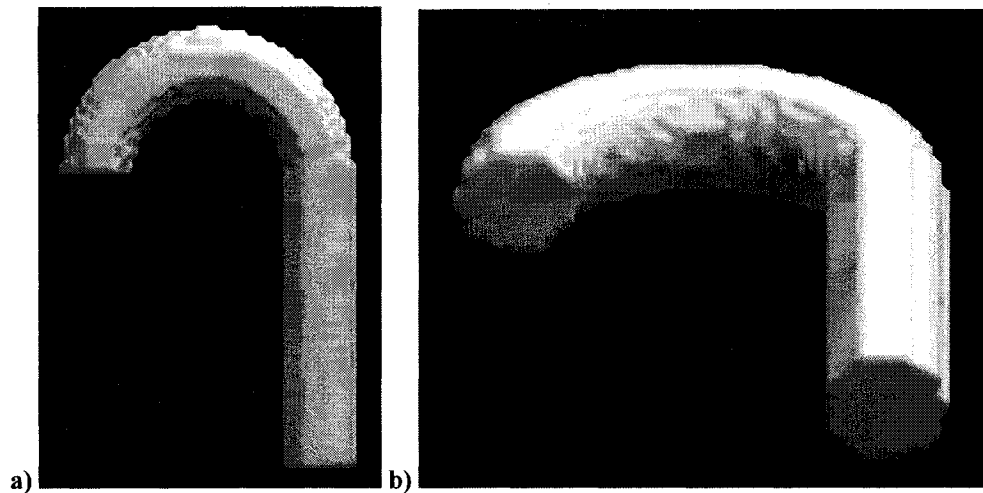
où  $r$  représente le rayon du cylindre (et du tore),  $L$  représente la longueur du cylindre et  $a$  représente le rayon de courbure du tore (voir la figure 4.9). La formule de l'aire se décompose en trois parties : la première concerne l'aire de la surface du cylindre, la seconde l'aire de la surface du demi-tore et la troisième l'aire des deux cercles apposés

aux extrémités du cylindre et du demi-tore pour obtenir une surface fermée. La formule du volume est divisée en deux parties : la première concerne le volume du cylindre et la seconde le volume du demi-tore. Les dimensions des données analytiques sont  $r = 0,5$ ,  $L = 4$  et  $a = 1,5$ .



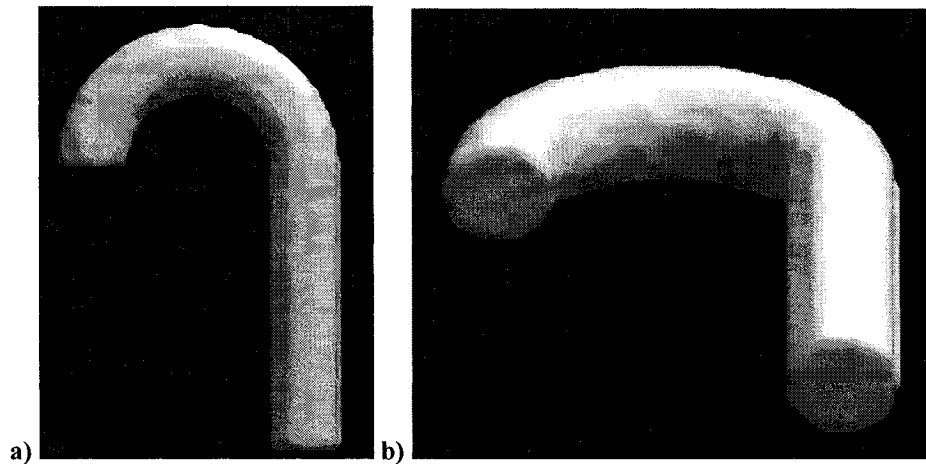
**Figure 4.9 : Dimensionnement des données analytiques.**

La figure 4.10 montre les surfaces obtenues pour cette série de données, produites directement sans traitement préalable avec une isovaleur de 0,5, la plage de valeurs possibles allant de 0,0 à 1,0.



**Figure 4.10 : Surfaces 3D pour les données analytiques, sans traitement.**

La figure 4.11 illustre les surfaces obtenues pour la série de données analytiques, produites avec une isovaleur de 0,5, suite à une séquence d'un seul traitement de filtre de moyenne pondérée (poids des voisins immédiats en  $i$ , en  $j$  et en  $k$  : 1; poids des autres voisins : 0).



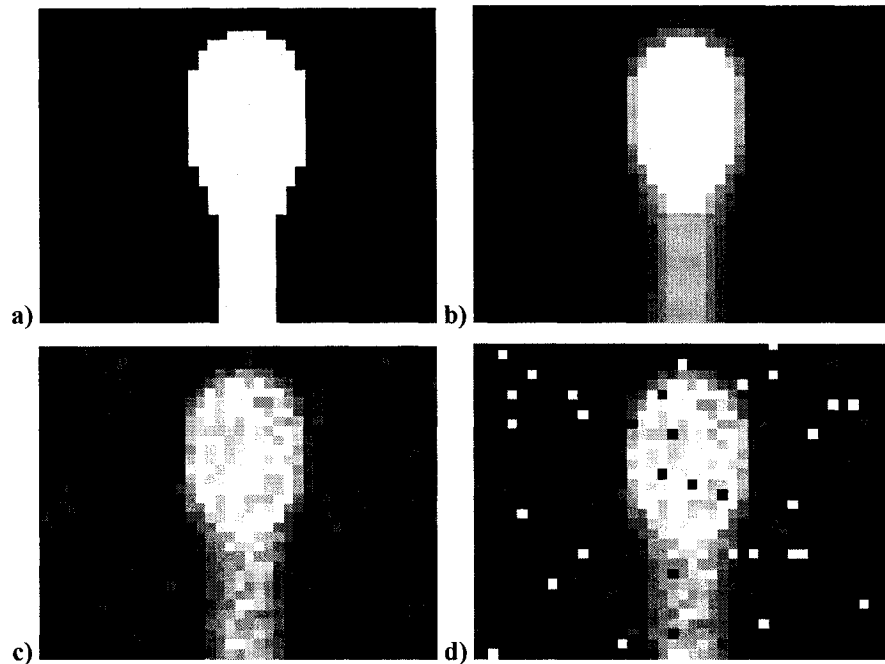
**Figure 4.11 : Surfaces 3D pour les données analytiques, après le filtre de moyenne pondérée.**

Afin de rendre les données analytiques un peu plus réalistes et avec certaines difficultés, nous les avons modifiées avec trois opérations. Nous avons d'abord appliqué le filtre de moyenne pondérée avec des poids de 1 pour les voisins immédiats en  $i$ ,  $j$  et  $k$  dans le but de rendre les frontières plus floues. Nous avons ensuite ajouté un bruit gaussien (le type de bruit dominant dans les images produites par IRM (Sutton, Noll et Fessler 2003)) de moyenne 0,0 et d'écart type 0,1 selon la formule suivante :

$$I_{finale}(i, j, k) = I_{initiale}(i, j, k) + b \quad (4.3)$$

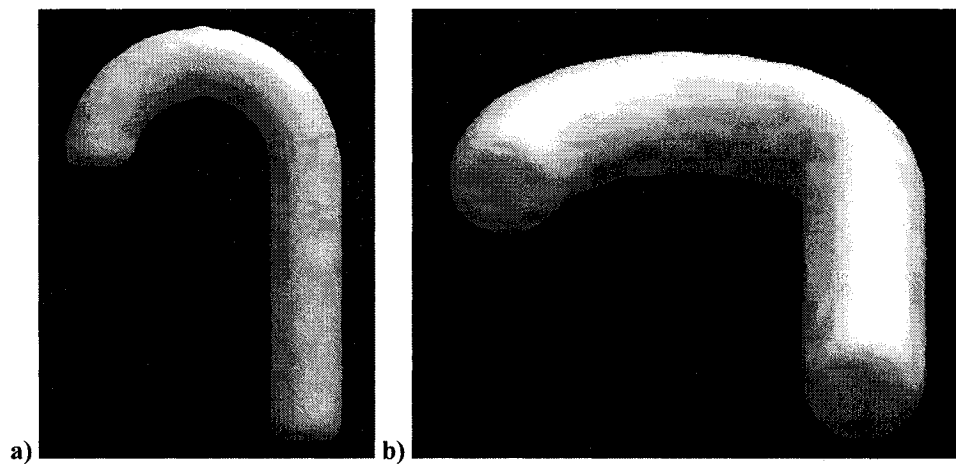
où  $b$  représente le bruit. Comme ce bruit, généré pseudo-aléatoirement, peut être négatif ou positif, l'intensité résultante doit être restreinte entre 0,0 et 1,0. Pour simuler des pertes d'informations, nous avons ajouté, en troisième étape, un bruit de style « sel et poivre » avec une probabilité de 0,05. Cela signifie que cinq pourcent des voxels, choisis spatialement de façon pseudo-aléatoire, se voient attribuer une nouvelle valeur, choisie pseudo-aléatoirement parmi les deux valeurs 0,0 et 1,0. Les valeurs numériques pour les deux types de bruit ont été choisies de manière à simuler visuellement le bruit présent

sur les images médicales sources utilisées dans ce mémoire. La figure 4.12 donne un exemple de cette application d'opérations.



**Figure 4.12 : Application de bruit sur les données analytiques. Données originales en a), données après le filtre de moyenne pondérée en b), données après l'ajout de bruit gaussien en c), données après l'ajout de bruit « sel et poivre » en d).**

La figure 4.13 donne les surfaces obtenues pour cette série de données analytiques bruitées, après l'application de la séquence de filtres présentée à la figure 4.14.



**Figure 4.13 : Surfaces 3D pour les données analytiques bruitées.**



- 1) Augmentation du contraste (seuils minimal et maximal : 0,3 et 0,5)
- 2) Filtre médian (noyau de taille 5 x 5 x 5)
- 3) Fermeture morphologique (élément structurant cubique de 3 x 3 x 3)
- 4) Filtre de moyenne pondérée (poids des voisins immédiats en i, en j et en k et poids des autres voisins : 1)
- 5) Croissance de région (germe aux coordonnées 64, 64, 84; seuils minimal et maximal : 0,2 et 1,0)
- 6) Augmentation du contraste (seuils minimal et maximal : 0,1 et 0,3)
- 7) Filtre de moyenne pondérée (poids des voisins immédiats en i, en j et en k et poids des autres voisins : 1)

**Figure 4.14 : Séquence de traitements pour la série données analytiques bruitées.**

Notons que la séquence de traitements appliquée est identique à celle de la série de la crosse aortique et des carotides (figure 4.2). La surface a été produite avec une isovaleur de 0,75, la plage de valeurs possibles allant de 0,0 à 1,0. Le tableau 4.3 donne les informations relatives à la surface obtenue et les informations réelles du modèle mathématique.

**Tableau 4.3 : Informations relatives à la surface pour la série données analytiques.**

	Données analytiques		Données analytiques bruitées	Modèle mathématique
	Sans traitement	Filtre de moyenne pondérée	Séquence de traitements	—
<b>Nombre de sommets</b>	6 182	5 046	6 173	—
<b>Nombre d'éléments (triangles)</b>	12 360	10 088	12 342	—
<b>Aire de la surface (mm<sup>2</sup>)</b>	30,1340	28,7325	28,4802	28,9352
<b>Volume interne de la surface (mm<sup>3</sup>)</b>	6,6628	6,5705	6,8336	6,8427

Le but d'utiliser des données analytiques est de pouvoir valider, dans une certaine mesure, les résultats obtenus. L'erreur sur l'aire de la surface et l'erreur sur le volume peuvent être calculées à partir des formules suivantes :

$$Erreur_{aire}(\%) = \frac{Aire_{réelle} - Aire_{reconstruite}}{Aire_{réelle}} \times 100 \quad (4.4)$$

$$Erreur_{volume}(\%) = \frac{Volume_{réel} - Volume_{reconstruit}}{Volume_{réel}} \times 100 \quad (4.5)$$

où Aire<sub>réelle</sub> et Volume<sub>réel</sub> représentent respectivement l'aire et le volume de la surface du modèle mathématique et Aire<sub>reconstruite</sub> et Volume<sub>reconstruit</sub> l'aire et le volume de la surface produite par notre système. Le tableau 4.4 regroupe les résultats de ces métriques de validation pour la série de données analytiques.

**Tableau 4.4 : Erreurs sur l'aire et le volume pour la série données analytiques.**

	Données analytiques		Données analytiques bruitées
	Sans traitement	Filtre de moyenne pondérée	Séquence de traitements
<b>Erreur sur l'aire</b>	- 4,1 %	0,7 %	1,6 %
<b>Erreur sur le volume</b>	2,6 %	4,0 %	0,1 %

#### 4.4 Temps d'exécution

Le programme a été testé sur un ordinateur personnel compatible Pentium 4 cadencé à 2,66 GHz avec 768 Mo de mémoire vive, sous le système d'exploitation Windows XP. Le tableau 4.5 regroupe les temps moyens d'exécution des algorithmes des traitements présentés au chapitre 2 pour les trois séries de données, de même que les écarts types des échantillons. Ces statistiques sont présentées à titre indicatif et ne constituent pas une étude exhaustive. Une dizaine de valeurs de temps pour des conditions typiques a été utilisée pour chaque calcul. Le cas #3 pour la série de la crosse aortique et des carotides a été laissé de côté car le nombre de voxels des données diffère des deux premiers cas.

**Tableau 4.5 : Temps moyens d'exécution des algorithmes des traitements et écarts types pour les trois séries de données (temps pour les séries complètes et temps par Mvoxels (1 000 000 voxels)).**

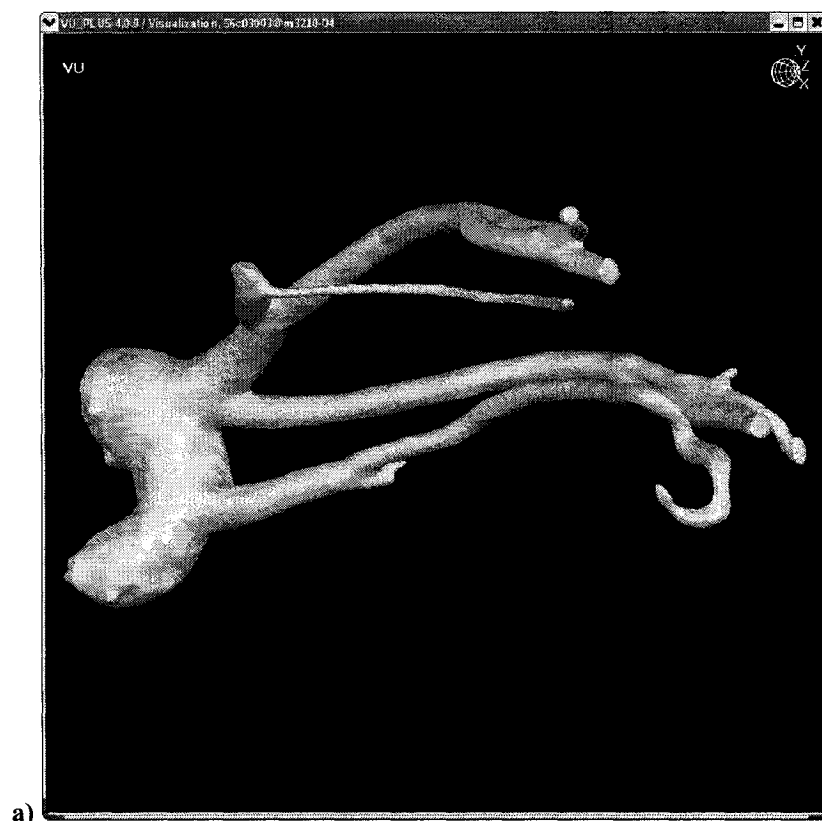
Traitement	Temps d'exécution (s)					
	Série crosse aortique et carotides (140x256x256 voxels)		Série aorte (5x256x256 voxels)		Série données analytiques (128x128x128 voxels)	
	Pour la série complète	Par Mvoxels	Pour la série complète	Par Mvoxels	Pour la série complète	Par Mvoxels
<b>Augmentation du contraste</b>	$\bar{x} = 1,964$ $s = 0,011$	0,214	$\bar{x} = 0,070$ $s = 0,001$	0,214	$\bar{x} = 0,451$ $s = 0,003$	0,214
<b>Filtre de moyenne pondérée</b>	$\bar{x} = 20,477$ $s = 0,194$	2,232	$\bar{x} = 0,640$ $s = 0,003$	1,952	$\bar{x} = 4,633$ $s = 0,034$	2,209
<b>Filtre médian (5x5x5)</b>	$\bar{x} = 80,337$ $s = 2,632$	8,756	—	—	$\bar{x} = 17,895$ $s = 0,060$	8,533
<b>Opération morphologique (dilatation)</b>	$\bar{x} = 38,015$ $s = 0,598$	4,143	—	—	$\bar{x} = 8,368$ $s = 0,030$	3,990
<b>Opération morphologique (érosion)</b>	$\bar{x} = 38,421$ $s = 0,303$	4,188	—	—	$\bar{x} = 8,523$ $s = 0,018$	4,067
<b>Croissance de région</b>	$\bar{x} = 6,343$ $s = 0,282$	0,691	$\bar{x} = 0,217$ $s = 0,005$	0,661	$\bar{x} = 1,438$ $s = 0,019$	0,686

$\bar{x}$  = moyenne arithmétique

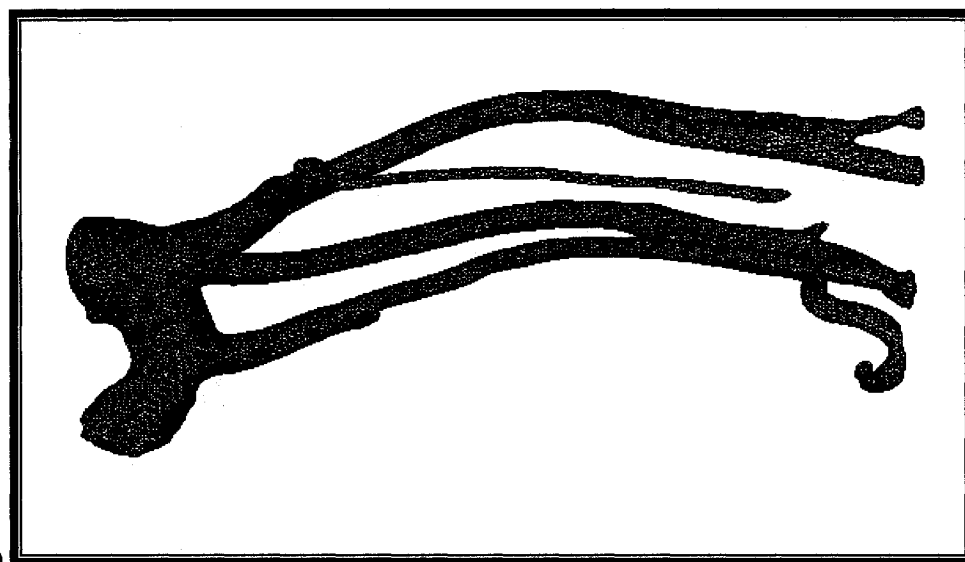
s = écart type de l'échantillon

## 4.5 Exportation des surfaces en formats « VU » et « YAMS »

La figure 4.15 montre l'exportation de la surface issue de la série de la crosse aortique et des carotides (cas #1) en formats « VU » et « YAMS », visualisée par les logiciels des formats respectifs.



a)



b)

Figure 4.15 : Exportation de la surface issue de la série crosse aortique et carotides en formats « VU » (a) et « YAMS » (b).

-----

Dans ce chapitre, nous avons montré les résultats produits par notre système à l'aide de l'application spécifique de la modélisation 3D de vaisseaux sanguins à partir d'images médicales. Nous avons présenté, pour trois séries de données différentes (images médicales de la crosse aortique et des carotides, images médicales de l'aorte, données analytiques représentant la crosse aortique et l'aorte), les surfaces 3D résultantes, les séquences de traitements appliquées sur les données sources pour produire les surfaces, et les mesures quantitatives d'aire et de volume lorsque applicables. Les temps moyens d'exécution ont été donnés pour chaque traitement. Le troisième objectif du mémoire, qui était de proposer des séquences de traitements permettant de retrouver une surface tridimensionnelle pour quelques groupes de données sources types, a été atteint.

Le chapitre suivant propose une analyse approfondie des résultats présentés dans ce chapitre et de l'ensemble du projet.

## **CHAPITRE 5 : DISCUSSION**

Ce chapitre discute les résultats présentés au chapitre précédent en revenant, pour chaque série de données, sur les séquences de traitements utilisées, les surfaces produites et la possibilité de validation. Ce chapitre vise également à commenter la méthodologie retenue en s'attardant sur certains aspects particuliers tels que les métriques de validation et l'approche uniformisante. En conclusion de ce chapitre, l'hypothèse de départ et les objectifs du projet sont revisités.

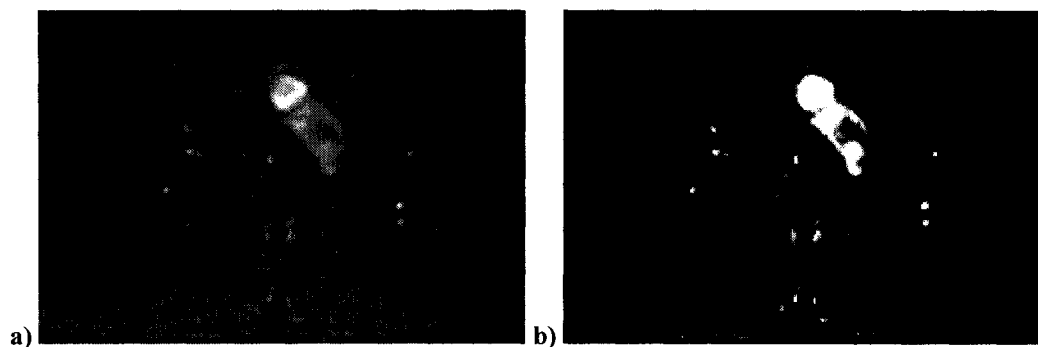
### **5.1 Discussion sur la série crosse aortique et carotides**

La série crosse aortique et carotides est sans doute la plus représentative, parmi les séries de données présentées au chapitre précédent, des volumes d'images médicales utilisés pour la reconstruction de modèles géométriques. Elle pose des difficultés importantes qui varient d'une section à l'autre de la série, comportant 140 images. Les sections suivantes examinent les trois cas de la série crosse aortique et carotides proposés à la section 4.1.

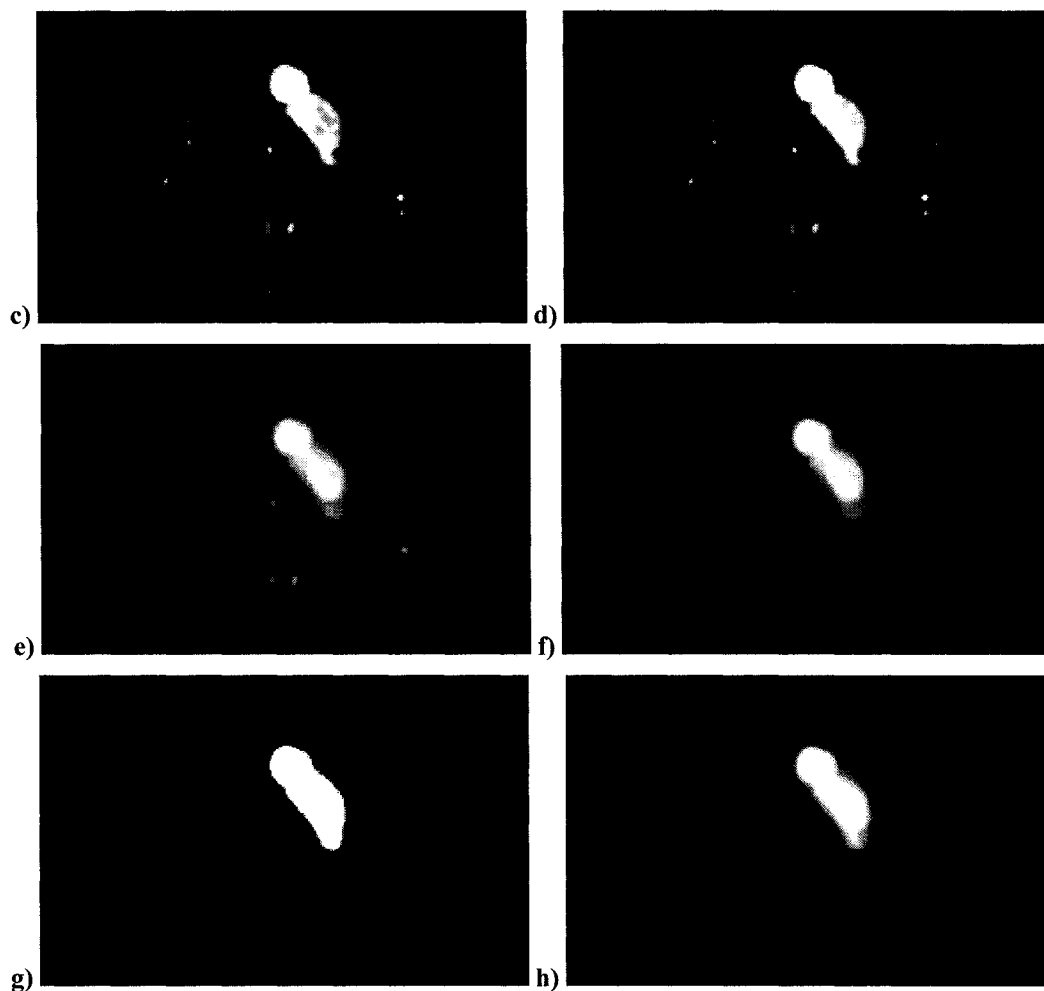
#### **5.1.1 Séquences de traitements et surfaces résultantes**

Le premier cas concerne un but de modélisation générale des vaisseaux sanguins de la série complète : on cherche à obtenir une surface acceptable pour toutes les parties. La figure 5.1 présente l'évolution des données après l'application en 3D de chacun des traitements à l'aide d'une image de la crosse aortique en coupe axiale (image numéro 18 sur 140). Cette coupe 2D a été choisie pour des fins de discussion et les autres coupes montrent des résultats semblables. Selon la figure 4.2 présentant la séquence de traitements utilisée, le premier traitement appliqué est une augmentation du contraste dans le but d'amplifier les limites entre les différents objets des données. Les seuils

relativement bas permettent du même coup d'éliminer les éléments foncés (particulièrement les autres organes) et quelques trous (régions d'intensité faible) présents dans les vaisseaux sanguins en augmentant au maximum l'intensité des voxels supérieurs à 0,5 (figure 5.1b). Par contre, quelques nouveaux trous apparaissent dans les régions où l'intensité des voxels était très faible. La seconde étape est l'application du filtre médian, visant à limiter le bruit impulsif et contrer l'effet négatif de la première étape (figure 5.1c). En troisième lieu, l'opération de fermeture morphologique (dilatation puis érosion) est exécutée dans le but d'éliminer les petites sections d'intensités trop faibles tout en conservant la taille des objets (figure 5.1d). Par la suite, les contours des données sont adoucis avec le filtre de moyenne pondérée de poids égaux (figure 5.1e). Le processus de segmentation est concrétisé avec l'application de l'algorithme de croissance de région en cinquième étape, qui élimine les objets non connectés avec un effet régulateur selon un seuil minimal assez bas car les différents objets sont déjà relativement bien délimités par rapport à l'arrière-plan nul (noir) (figure 5.1f). L'augmentation du contraste qui suit est utilisée dans un but différent de celui de la première étape : augmenter la taille de l'objet en prévision de la dernière étape de lissage, nécessaire pour obtenir une isosurface lisse, qui a tendance à rapetisser les objets (figure 5.1g). Les seuils ont été choisis pour encadrer symétriquement l'intensité limite conservée par la croissance de région. Enfin, un lissage à l'aide du filtre de moyenne pondérée est appliqué pour que la surface produite soit d'allure lisse (figure 5.1h).



**Figure 5.1 : Effets de la séquence de traitements sur la série crosse aortique et carotides (cas #1), étape par étape pour une image en coupe axiale (image numéro 18 sur 140). En a), données originales; en b), augmentation du contraste.**



**Figure 5.1 (suite) : En c), Filtre médian; en d), fermeture morphologique; en e), filtre de moyenne pondérée; en f), croissance de région; en g), augmentation du contraste; en h), filtre de moyenne pondérée.**

En considérant les figures 5.1a et 5.1h, nous pouvons constater que dans le plan 2D, la séquence d'opérations a rendu possible la délimitation d'un objet plein tout en conservant son allure initiale. Cette tranche dans le plan image est davantage en accord avec les images environnantes grâce aux traitements appliqués en 3D, qui ont permis de profiter de la cohérence 3D des données sources. En observant la figure 4.1 montrant la surface 3D résultante pour le premier cas, on peut remarquer que celle-ci est d'allure lisse et généralement en accord avec l'anatomie humaine. En effet, la crosse aortique et les principales branches des carotides sont récupérées, sans plus. Toutefois, certaines



sections des carotides ont été faussement fusionnées (voir la figure 4.1d). Cela est attribuable au fait que pour obtenir une crosse aortique lisse et complète, beaucoup d'opérations de filtrage ayant tendance à propager les valeurs d'intensités voisines en trois dimensions ont été nécessaires. Puisque ces sections des carotides se trouvent très près les unes des autres, elles ont été reliées. Afin de retrouver une surface avec les branches des carotides séparées, il faudrait donc appliquer moins de traitements modifiant les voxels basés sur leur concordance avec leur voisinage et être plus sélectif quant aux plages d'intensités tolérées tel que dans le cas suivant.

Le second cas présenté à la section 4.1 vise un but de modélisation avec une attention particulière portée aux carotides : il est primordial d'obtenir les branches des carotides séparées les unes des autres. Selon la figure 4.4 présentant la séquence de traitements utilisée, le premier traitement exécuté est une augmentation du contraste avec des seuils légèrement plus sévères (élevés) que pour le premier cas, dans le but d'accentuer la séparation des carotides. Cela a pour conséquence de laisser et de créer d'importants trous dans la région de la crosse aortique. Cette opération est immédiatement suivie par la croissance de région qui régularise légèrement la région de la crosse aortique et élimine les autres objets non connectés. Enfin, une opération de lissage à l'aide du filtre de moyenne pondérée est exécutée en accordant un poids valide seulement aux voxels immédiatement voisins afin d'adoucir les frontières tout en conservant la séparation des carotides.

La figure 4.3 (surface 3D résultante) illustre la conséquence de la sévérité des traitements appliqués : toutes les branches des carotides sont récupérées et sont très bien séparées les unes des autres, au détriment toutefois de l'allure de la crosse aortique qui présente une surface parsemée de trous. Les opérations cherchant à boucher des trous et éliminer le bruit impulsif comme la fermeture morphologique et le filtre médian ne peuvent être utilisées dans ce cas. En effet, dans l'exemple de la fermeture, l'étape de dilatation aurait nécessairement comme conséquence de relier les carotides.

Enfin, le troisième cas, provenant de la série tronquée de la crosse aortique et des carotides et traité au chapitre précédent, met l'accent sur la modélisation de la crosse aortique. La séquence de traitements proposée retient globalement la même structure que celle du premier cas, à quelques petites différences près. Selon la figure 4.6 présentant la séquence de traitements retenue, la première opération concerne le filtre médian, appliqué deux fois dans le but d'éliminer le plus possible les trous présents dans la structure de la crosse aortique. La taille du filtre de  $5^3$  voxels offre la possibilité de donner aux voxels une cohérence avec leur environnement à plus grande échelle que pour une taille de  $3^3$ , sans nécessiter un temps de calcul très imposant. Le traitement suivant, la fermeture morphologique, est exécuté comme dans le premier cas dans le but d'éliminer les petites sections d'intensités trop faibles tout en conservant la taille des objets. Il est également suivi du filtre de moyenne pondérée, mais les poids accordés aux voxels voisins diffèrent : les voxels immédiatement voisins en  $k$ , c'est-à-dire ceux qui se trouvent dans les plans images précédent et suivant, se voient accorder plus d'importance afin d'améliorer la cohérence des plans images entre eux. Cela permet de remplir les trous indésirables. La cinquième opération concerne la segmentation avec la croissance de région, qui offre un léger effet régulateur et élimine les objets non connectés. En sixième lieu, à l'instar du premier cas, une augmentation du contraste est effectuée avec des seuils encadrant symétriquement le seuil minimal de l'étape précédente. Enfin, trois étapes de lissage avec le filtre de moyenne pondérée sont appliquées afin d'obtenir une surface d'allure lisse et plus près de sa taille originale.

La figure 4.5 illustrant la surface 3D résultante permet de constater que la séquence de traitements a produit une surface très lisse, plus lisse que dans le premier cas cherchant à modéliser globalement la crosse aortique et les carotides, et certainement plus réaliste que dans le second cas où de nombreux trous sont présents dans la crosse aortique. Cela s'explique par le choix des paramètres et de l'ordre des traitements qui n'a pas tenu compte de la petite taille des carotides. Il serait approprié d'envisager l'utilisation d'un filtre qui tienne compte des différences géométriques entre la crosse aortique et les

carotides. Par exemple, la sévérité du filtre pourrait dépendre de la courbure locale de l'objet : une sélection des seuils plus sévère pour une région de haute courbure (telles que les carotides) et une sélection moins sévère pour une région de moindre courbure (telle que la crosse aortique). La figure 4.5 nous permet également de constater que l'une des deux extrémités de la crosse semble légèrement trop étroite par rapport à la réalité, de même que dans les deux autres cas. Cela est attribuable au fait que les images sources ne présentent pas clairement l'information géométrique de cette partie d'organe. La figure 5.2 montre cette partie mal définie sur la première image des données sources de la série crosse aortique et carotides, en comparaison avec la sixième image de la série pour laquelle cette partie est bien définie. Pour retrouver une crosse aortique plus fidèle à ce niveau, on pourrait songer à l'incorporation d'informations connues a priori provenant d'un modèle générique de crosse aortique ou encore de l'évaluation des images par un spécialiste.

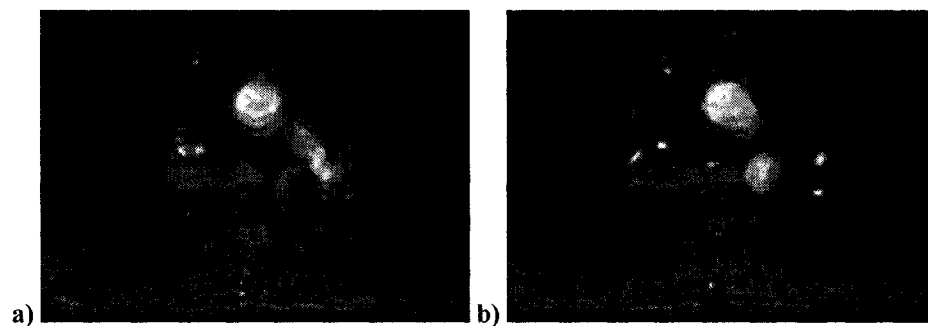


Figure 5.2 : Extrémité de la crosse aortique dans les données sources, mal définie en a) et bien définie en b) (voir flèches).

### 5.1.2 Validation

La validation de la modélisation de la surface 3D de la crosse aortique et des carotides à partir de données *in vivo* telles que les images médicales est une tâche ardue car il n'existe pas de gabarit de géométries *in vivo* permettant la comparaison quantitative des résultats (Moore, Steinman, Holdsworth et Ethier 1999). Il serait toutefois possible d'utiliser des gabarits *in vitro* comme des modèles de crosse aortique et de carotides

fabriqués puis imagés par résonance magnétique, dont les données géométriques sont connues. Ces gabarits *in vitro* permettraient de valider la méthodologie de modélisation, mais non de valider directement les surfaces 3D produites à partir des données *in vivo*. Comme nous n'avions pas de tels gabarits *in vitro* à notre disposition, nous avons opté pour l'utilisation de données analytiques représentant la crosse aortique et l'aorte (analysée à la section 5.3).

Afin d'apporter des éléments de validation pour les surfaces 3D produites à partir de données *in vivo*, nous avons employé les méthodes suivantes :

1. comparaison qualitative des images segmentées par notre logiciel (ayant servi à la création de la surface 3D finale) avec les images médicales sources;
2. comparaison qualitative de la surface 3D produite avec la géométrie *in vivo* (image médicale de l'ensemble de la crosse aortique et des carotides);
3. comparaisons qualitative et quantitative de la surface 3D produite avec le résultat d'une segmentation manuelle des mêmes données sources.

Bien que la première méthode de validation se fasse en 2D alors que les résultats sont tridimensionnels, elle offre la possibilité de faire une évaluation qualitative relativement facilement. La figure 5.3 montre diverses coupes des données originales et des données traitées correspondantes pour le premier cas étudié de la série de la crosse aortique et des carotides.

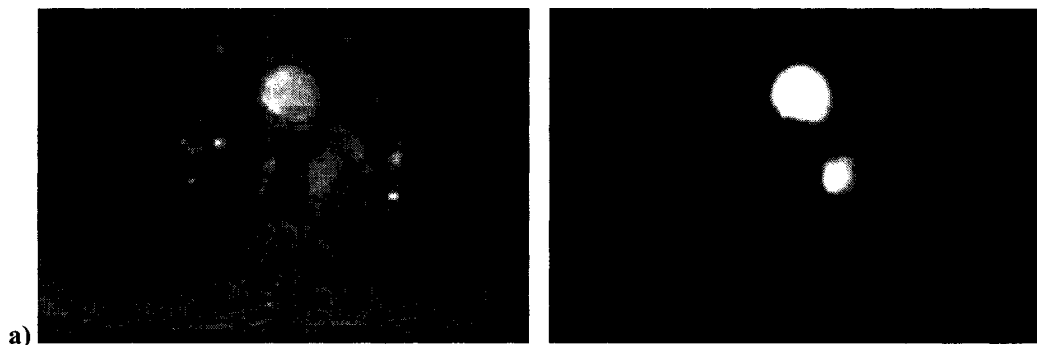
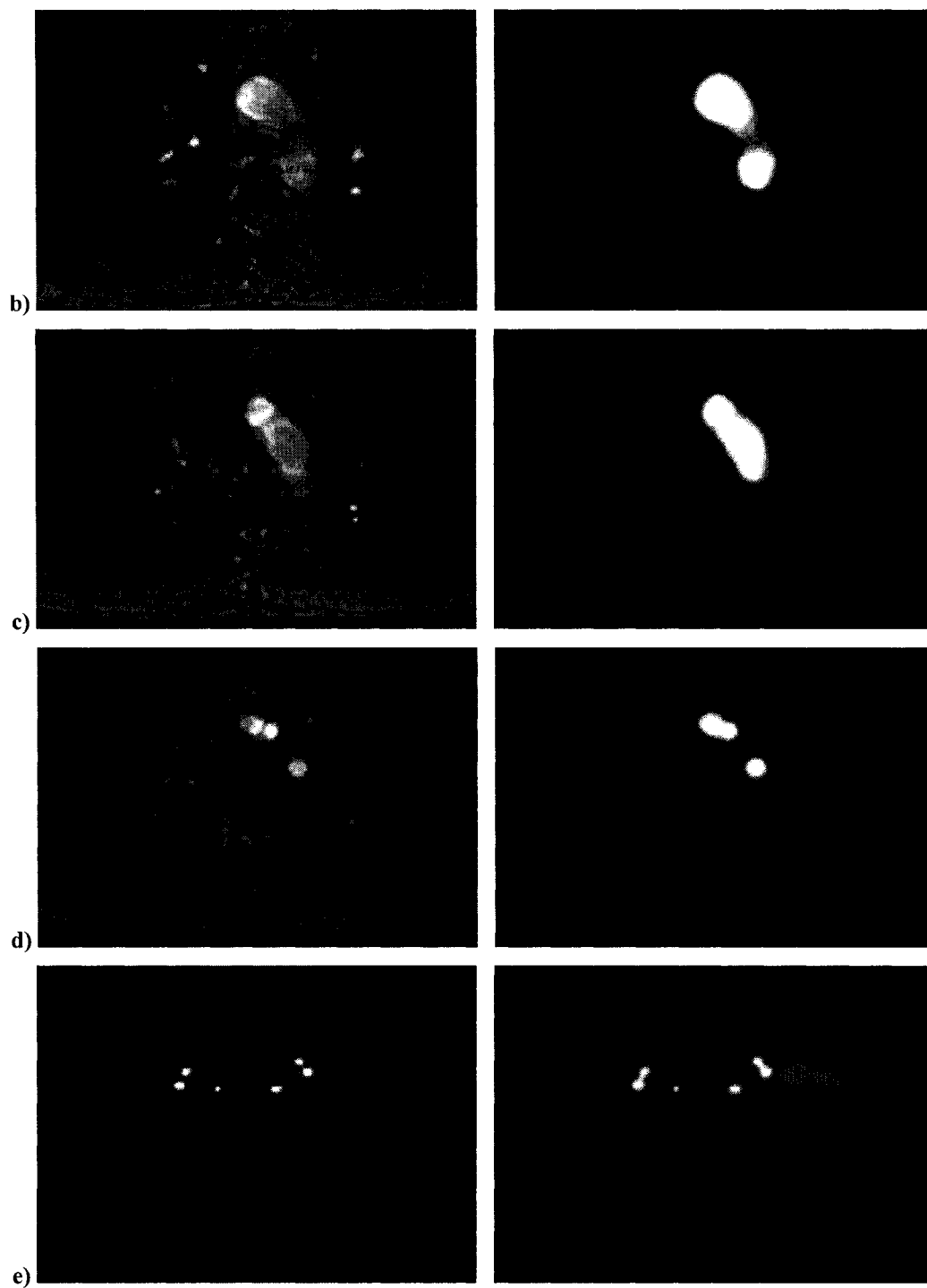
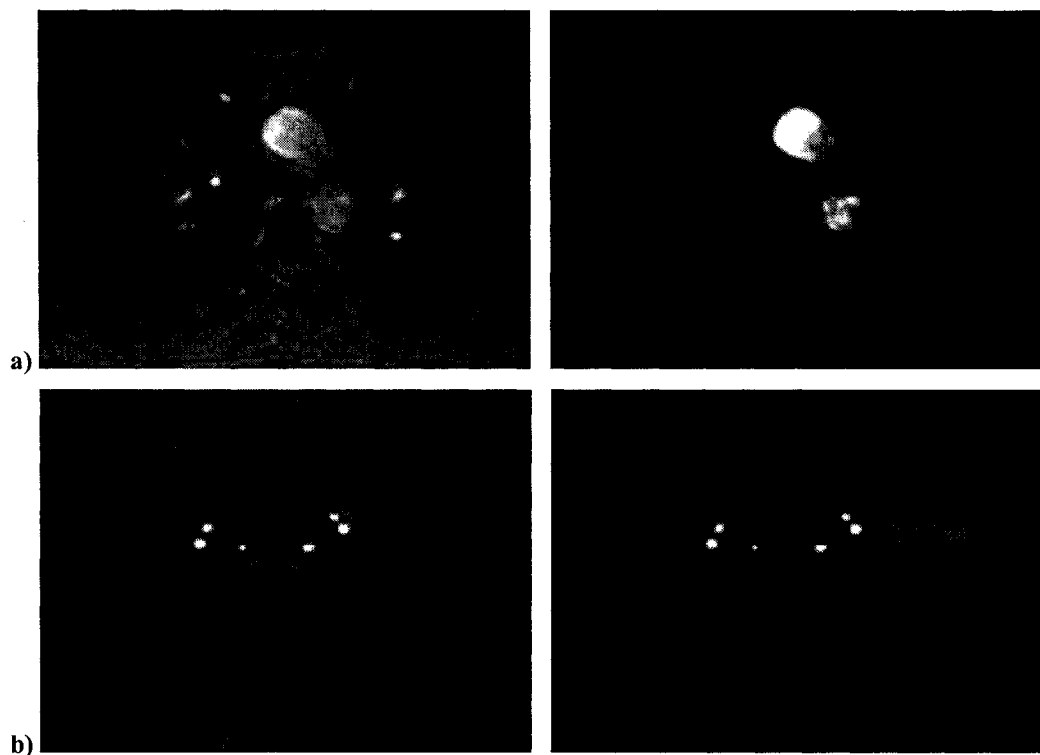


Figure 5.3 : Section des images originales (colonne de gauche) versus images segmentées (colonne de droite) pour le cas #1 de la série crosse aortique et carotides. En a), image numéro 3.



**Figure 5.3 (suite) : En b), image numéro 7; en c), image numéro 17; en d), image numéro 25; en e), image numéro 113.**

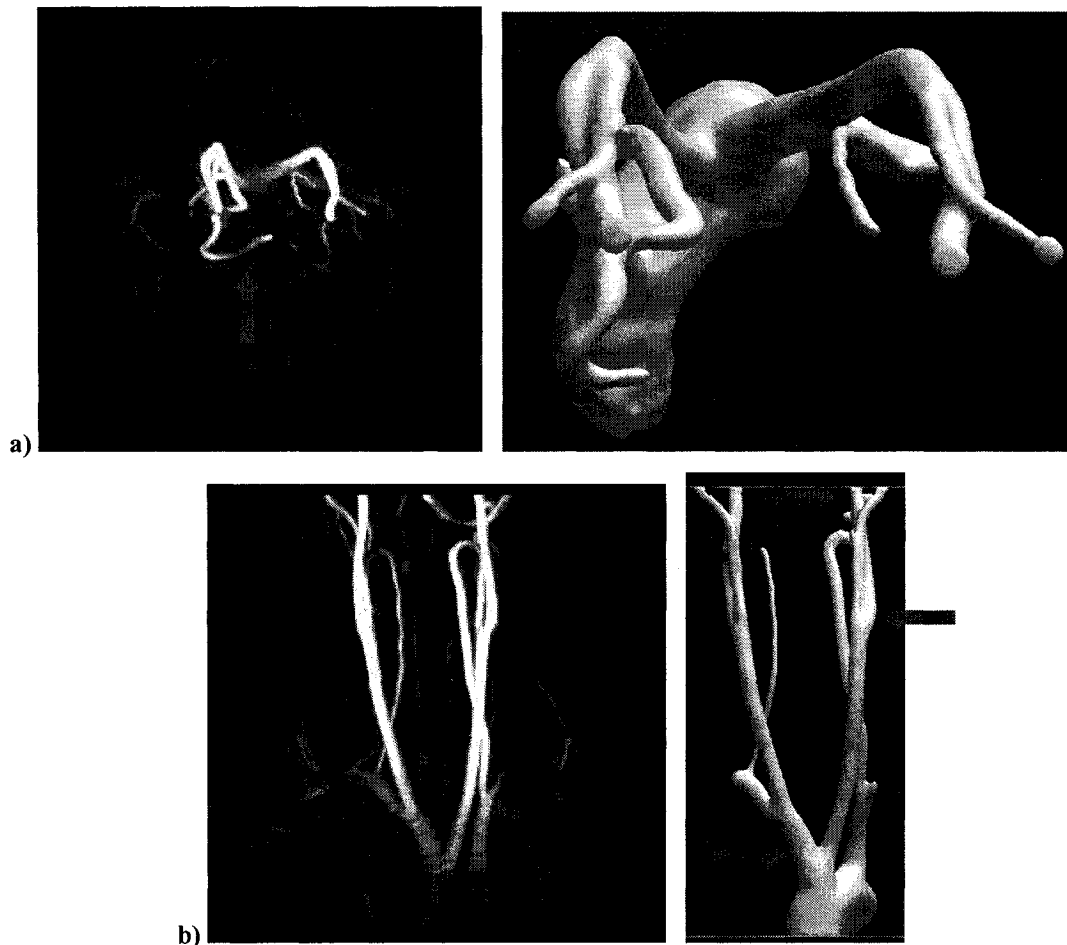
Les images de la figure 5.3 nous indiquent deux choses. La séquence de traitements appliquée permet de pallier aux artefacts des données sources car les objets retrouvés sont sans trou (a, b et c) et de taille et de géométrie globalement réalistes (a, b, c, d et e), mais elle a pour conséquence de fusionner certains objets très près les uns des autres, comme dans la zone de bifurcation des carotides (d) et les extrémités des carotides (e). La figure 5.4 présente les mêmes comparaisons pour le second cas étudié de la série de la crosse aortique et des carotides.



**Figure 5.4 : Sections des images originales (colonne de gauche) versus images segmentées (colonne de droite) pour le cas #2 de la série crosse aortique et carotides. En a), image numéro 7; en b), image numéro 113.**

Les images de la figure 5.4 nous indiquent encore une fois deux choses. La séquence de traitements appliquée ne permet pas de pallier totalement aux artefacts des données sources car les objets montrent des trous, c'est-à-dire des zones plus foncées (a), mais elle permet de récupérer correctement les objets très près les uns des autres, comme dans les extrémités des carotides (b).

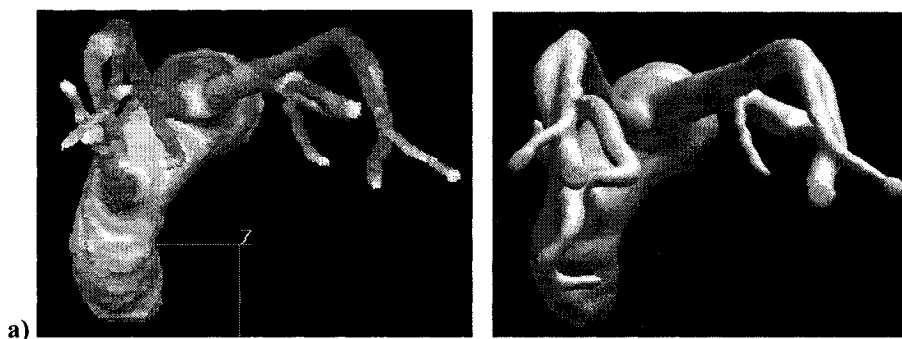
La seconde méthode de validation utilise d'autres images qui ont été produites sous différentes coupes et conditions lors de la séance d'acquisition de la série de 140 images médicales de la crosse aortique et des carotides. Certaines présentent un intérêt majeur pour la validation des résultats car elles illustrent d'un seul coup d'oeil l'ensemble de l'aorte et des carotides en vues axiale et coronale. Contrairement aux images de la série de 140 qui étaient très minces et peu espacées physiquement, ces images d'ensemble sont uniques à leur série et très épaisses (220 millimètres). Elles montrent donc la géométrie *in vivo* 3D complète. La figure 5.5 fournit les informations nécessaires pour la seconde méthode de comparaison, une comparaison qualitative de la surface 3D créée par notre logiciel avec la géométrie *in vivo*.



**Figure 5.5 : Géométrie *in vivo* (images de gauche) versus surface reconstruite par notre logiciel (images de droite) pour le cas #1 de la série crosse aortique et carotides. En a), vue axiale; en b), vue coronale.**

Les images de la figure 5.5 confirment les observations mentionnées précédemment au sujet de la performance de la séquence de traitements : la séquence permet de pallier aux artefacts des données sources car la surface retrouvée est exempte de trou et de taille et de géométrie globalement réalistes, mais elle a pour conséquence de fusionner certains objets très près les uns des autres, comme dans la zone de bifurcation des carotides où la séparation se fait légèrement un peu trop haut, et les extrémités des carotides (voir les flèches du bas et du milieu sur la figure b). Ces images ajoutent de nouvelles informations menant à l'évaluation de la séquence de traitements : on peut constater qu'une section de la carotide interne de gauche est manquante (voir la flèche du haut sur la figure b et celle sur la figure a) et que certaines régions ont été terminées un peu hâtivement (par exemple les artères sous-clavières). Ces régions manquantes présentent des intensités plus faibles que les autres régions et il est correct de penser qu'il serait possible de les récupérer par des valeurs de paramètres différentes (moins sévères) pour les traitements appliqués.

La troisième méthode de validation proposée concerne les comparaisons qualitative et quantitative de la surface 3D produite avec le résultat d'une segmentation manuelle des mêmes données sources (Lincourt-Éthier 2005, pages 38–40). La segmentation manuelle a été effectuée en deux dimensions, image par image, puis les segments 2D ont été par la suite assemblés en un modèle 3D. La figure 5.6 montre le résultat visuel de la segmentation manuelle en comparaison des surfaces retrouvées par notre logiciel.



**Figure 5.6 : Segmentation manuelle (images de gauche) versus surface reconstruite par notre logiciel (images de droite) pour le cas #1 de la série crosse aortique et carotides. En a), vue axiale.**



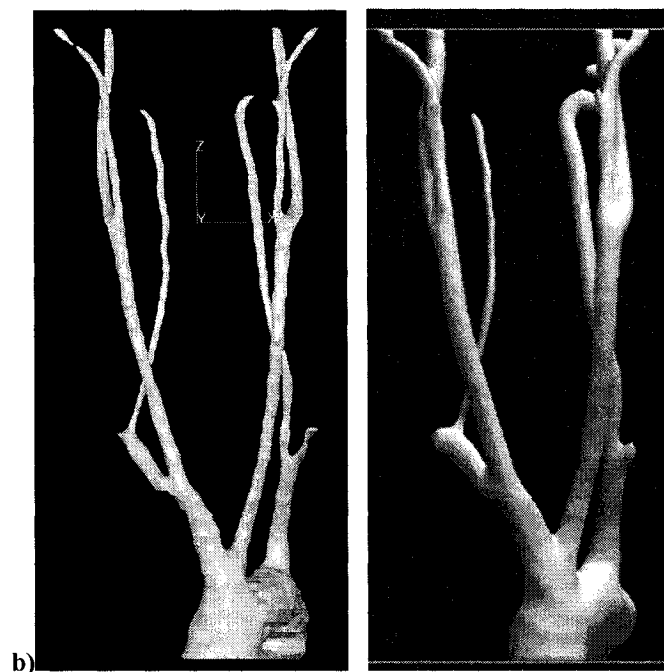


Figure 5.6 (suite) : En b), vue coronale.

En considérant la figure 5.6, nous pouvons remarquer que la surface retrouvée à partir des séquences de traitements semble être plus volumineuse que celle produite par segmentation manuelle : le diamètre des carotides est légèrement plus gros. Les extrémités de la crosse aortique sont davantage conformes au modèle générique d'une crosse aortique dans le cas de la segmentation manuelle car l'opérateur humain a pu incorporer son expertise dans la segmentation. Par contre, la surface issue de la segmentation manuelle montre une carotide interne gauche tronquée plus hâtivement que pour la surface retrouvée par notre logiciel. Cela confirme le fait qu'il est ardu de faire une segmentation manuelle en deux dimensions, image par image, pour une série volumineuse, de manière à ce que les segments sur chaque image soient cohérents avec leur environnement tridimensionnel, et que le résultat présente un certain côté arbitraire dû à l'obligation imposée à l'utilisateur de s'assurer manuellement de la cohérence spatiale des données d'une image à l'autre. L'utilisation de résultats de segmentation manuelle dans un but de validation s'en trouve limitée.

Nous avons tout de même fait l'exercice de mesurer l'aire et le volume interne de la surface issue de la segmentation manuelle dans le but de les comparer aux mesures présentées précédemment au tableau 4.1. L'aire de la surface issue de la segmentation manuelle est de  $15\,181\text{ mm}^2$  et son volume interne est de  $32\,829\text{ mm}^3$ . En se basant sur les formules d'erreur présentées au chapitre 4 (équations 4.4 et 4.5), l'erreur sur l'aire est de -43,3% et de -68,6% sur le volume pour le premier cas. Le signe négatif signifie que l'aire et le volume interne de la surface reconstruite par notre logiciel sont supérieurs d'environ 43% et 69%. Ces nombres très élevés, malgré la ressemblance qualitative des résultats, sont difficiles à évaluer. En faisant le même exercice pour le second cas (carotides bien séparées mais crosse aortique trouée), on retrouve une erreur sur l'aire de -18,2% et de -9,4% sur le volume. Ces erreurs sont beaucoup plus acceptables que celles du premier cas, malgré le fait que la surface semble visuellement moins bonne. La validation quantitative n'est pas concluante.

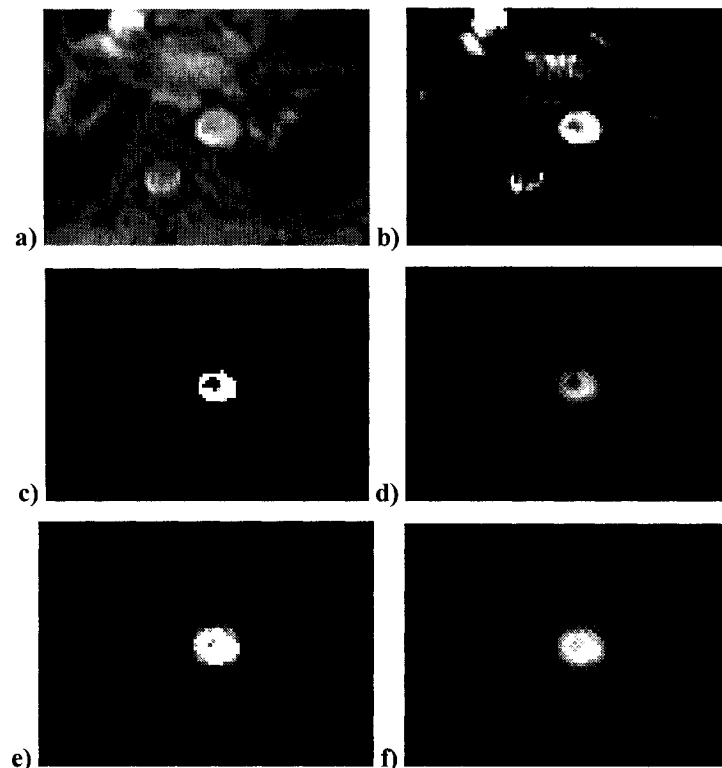
## **5.2 Discussion sur la série aorte**

La série aorte diffère de la série crosse aortique et carotides par le but de l'acquisition. Proposant un survol des organes, peu d'images (5) composent la série, et les voxels ont une forme de prisme rectangulaire. Cette série pose des difficultés quant à la récupération des objets désirés et non désirés et à l'élimination d'artéfacts. Les sections suivantes examinent les résultats de la série aorte exposés à la section 4.2.

### **5.2.1 Séquences de traitements et surfaces résultantes**

La figure 5.7 présente l'évolution des données après l'application de chacun des traitements à l'aide d'une section d'image de l'aorte en coupe axiale. Selon la figure 4.8 présentant la séquence de traitements utilisée, le premier traitement appliqué est une augmentation du contraste avec des seuils assez élevés dans le but d'éliminer le plus

d'objets non désirés possibles (figure b). Le second traitement consiste à segmenter les données à l'aide de la croissance de région, avec un seuil minimal très élevé pour éviter de récupérer une région indésirable connectée à l'aorte par un « pont » d'intensité moyenne. Le trou présent sur l'image 5 (figure c) est trop gros pour être éliminé par l'effet régulateur de la croissance de région. La troisième étape concerne l'application consécutive de deux filtres de moyenne pondérée, qui tentent de remplir les trous, considérant seulement les voxels voisins dans le plan image car les images précédente et suivante sont trop éloignées physiquement de l'image courante (figure d). Suite à ces lissages, les objets sont rendus plus petits que dans la réalité, c'est pourquoi leur taille est restaurée avec l'augmentation du contraste qui suit (figure e). Enfin, une dernière étape de lissage à l'aide du filtre de moyenne pondérée est nécessaire afin d'adoucir les frontières des objets (figure f).



**Figure 5.7 : Effets de la séquence de traitements sur la série aorte étape par étape pour une section de l'image 5 en coupe axiale. En a), données originales; en b), augmentation du contraste; en c), croissance de région; en d), 2x filtre de moyenne pondérée; en e), augmentation du contraste; en f), filtre de moyenne pondérée.**

Peu de traitements différents ont été utilisés dans la séquence. À cause de l'anisotropie prononcée de la forme des voxels, les traitements considérant des voisinages 3D tels que le filtre médian et les opérations morphologiques ont été proscrits pour ne pas déformer les sections de l'aorte. La figure 4.7 présentant la surface obtenue sous divers angles permet de constater que la séquence de traitements a produit une surface d'allure généralement correcte mais avec quelques régions rugueuses. Une façon de pallier à ce problème serait de rendre les voxels isotropes en intercalant des images créées par interpolation entre les images existantes.

### 5.2.2 Validation

Tout comme pour la série de la crosse aortique et des carotides, la validation de la modélisation de la surface 3D de l'aorte à partir de données *in vivo* telles que les images médicales est difficile à cause de l'absence de gabarit de géométries *in vivo* permettant la comparaison quantitative des résultats. Le seul élément de validation possible consiste à comparer qualitativement des images segmentées par notre logiciel (ayant servi à la création de la surface 3D finale) avec les images médicales sources. La figure 5.8 montre une section des cinq images de la série en versions originale et traitée.

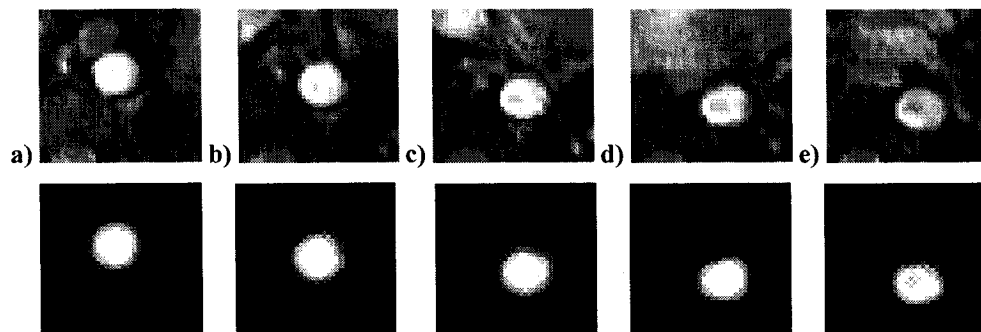


Figure 5.8 : Sections des images originales (rangée du haut) versus images segmentées (rangée du bas) pour la série aorte. De a) à e), images numéros 1 à 5.

Les images de la figure 5.8 permettent de constater que la séquence de traitements appliquée a eu pour effet de régulariser l'intensité de l'aorte et donc de retrouver une seule surface, sans trou indésirable. La géométrie de l'aorte semble bien conservée.

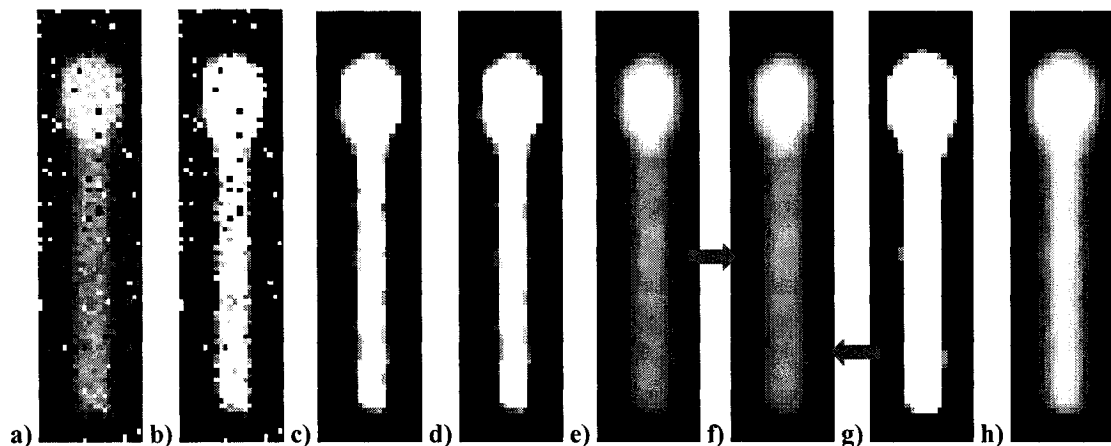
### 5.3 Discussion sur les données analytiques

Les données analytiques sont utilisées dans une optique de validation quantitative. Créées à partir d'un modèle mathématique, elles peuvent être considérées comme un gabarit *in vitro* de la crosse aortique et de l'aorte. Toutefois, puisque le modèle n'a pas été construit puis imagé par un scanneur, les artéfacts dus au processus d'acquisition d'images médicales sont absents. La reconstruction de la surface à partir des données analytiques telles quelles ne posent pas de difficulté car les frontières de l'objet sont bien définies. Nous avons donc tenté de simuler du bruit dans les données en appliquant en premier lieu un opérateur de lissage puis en ajoutant un bruit gaussien et un bruit de type « sel et poivre ». Les sections suivantes examinent les résultats de la série de données analytiques composée de  $128 \times 128 \times 128$  voxels, exposés à la section 4.3.

#### 5.3.1 Séquences de traitements et surfaces résultantes

Dans un premier temps, la surface de la crosse aortique et de l'aorte a été produite sans appliquer de traitement, directement à partir des données analytiques originales (non bruitées) afin de vérifier les erreurs engendrées par le processus de discrétisation du modèle mathématique et de création de l'isosurface. La figure 4.10, montrant la surface 3D pour les données analytiques sans traitement, nous permet de remarquer que la surface résultante n'est pas lisse et comporte une asymétrie au niveau de la section droite de l'aorte présente dans les données sources. Le fait d'appliquer le filtre de moyenne pondérée avant de créer la surface permet d'obtenir une surface plus lisse, mais avec la même asymétrie (voir la figure 4.11).

La séquence de traitements utilisée pour les données analytiques bruitées (figure 4.14) est la même que celle pour le premier cas étudié de la série crosse aortique et carotides. L'effet de chacun des traitements est semblable et est illustré à la figure 5.9 à l'aide de sections d'images dans le plan sagittal.



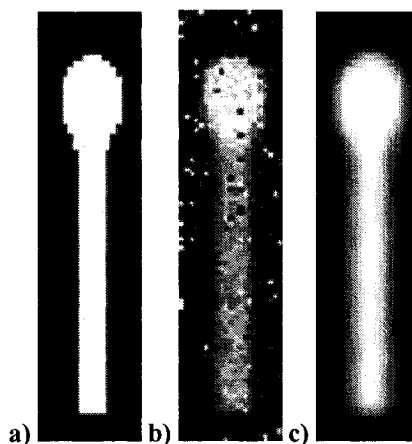
**Figure 5.9 : Effets de la séquence de traitements sur les données analytiques bruitées, étape par étape, pour une section d'image en coupe sagittale (image numéro 78). En a), données originales; en b), augmentation du contraste; en c), filtre médian; en d), fermeture morphologique; en e), filtre de moyenne pondérée; en f), croissance de région; en g), augmentation du contraste; en h), filtre de moyenne pondérée.**

Les images de la figure 5.9 nous indiquent que la séquence de traitements utilisée a été généralement efficace pour éliminer le bruit présent sur les données sources et pour produire un objet d'intensité uniforme et aux frontières relativement lisses. Le seuil minimal très bas de la croissance de région a toutefois récupéré deux petites régions indésirables (voir flèches sur figure f), qui ont été amplifiées par l'augmentation du contraste subséquente (figure g). Il aurait été possible de corriger ce problème en choisissant un seuil légèrement plus élevé pour la croissance de région, mais nous avons tenu à utiliser la séquence de traitements identique à celle du premier cas de la série crosse aortique et carotides dans un but de validation. La figure 4.13 présentant la surface obtenue permet de constater que la séquence de traitements a permis de produire une surface d'allure généralement correcte et lisse, beaucoup plus lisse que la surface créée directement à partir des données analytiques originales, mais avec des extrémités trop arrondies. Cet arrondissement est attribuable aux nombreux traitements de lissage.

Une façon de prévenir ce problème serait d'explorer l'utilisation d'une méthode de lissage telle que la diffusion anisotrope, qui combine le lissage de régions dont l'intensité varie lentement et la mise en valeur des frontières des objets (Gerig, Kübler, Kikinis et Jolesz 1992).

### 5.3.2 Validation

La validation de la modélisation de la surface 3D de la crosse aortique et de l'aorte à partir de données analytiques peut se faire qualitativement, en comparant des images segmentées par notre logiciel (ayant servi à la création de la surface 3D finale) avec les données sources, et quantitativement car l'aire et le volume réels du modèle mathématique discrétisé pour forger les données sources sont connus. Puisque la même séquence de traitements a été utilisée pour les données analytiques que pour le premier cas de la série crosse aortique et carotides, la validation quantitative présentée dans cette section fournit également un élément de validation pour les surfaces créées à partir d'images médicales *in vivo*. La figure 5.10 montre des sections d'une image dans le plan sagittal de la série en versions originale, bruitée et traitée.



**Figure 5.10 : Sections de l'image numéro 78 originale (a) versus image bruitée (b) versus image segmentée (c) pour la série données analytiques en coupe sagittale.**

Considérant que la surface résultante a été créée à partir des données traitées selon une isovaleur de 0,5, la taille globale de l'objet semble bien conservée mais les frontières ont été trop arrondies. En étudiant le tableau 4.4 présentant les erreurs sur l'aire et le volume, on peut constater que le fait de créer une isosurface directement à partir des données originales cause des erreurs quantitatives plus importantes (erreurs sur l'aire de -4,1 % et sur le volume de 2,6 %) que lorsque la surface est créée à partir des données bruitées à la suite de la séquence de traitements proposée (erreurs sur l'aire de 1,6 % et sur le volume de 0,1 %). Le fait de créer la surface à partir des données originales après l'application du filtre de moyenne pondérée diminue l'erreur sur l'aire (0,7 %) mais augmente l'erreur sur le volume (4,0 %). Nous pouvons en tirer deux choses : le processus de discrétisation introduit effectivement des erreurs qu'il est possible corriger partiellement avec l'application de traitements; l'utilisation de métriques globales comme l'aire et le volume interne de la surface ne permet pas de valider adéquatement les surfaces en totalité. En effet, bien que les erreurs soient très faibles, le fait que la surface produite suite à la séquence de traitements montre des extrémités trop arrondies laisse entrevoir la nécessité d'utiliser, en plus de métriques globales, des métriques agissant localement.

#### **5.4 Discussion sur les temps d'exécution et la complexité**

Les temps d'exécution des traitements implantés et utilisés dans ce projet ont été présentés au tableau 4.5 pour les trois séries de données : crosse aortique et carotides, aorte, et données analytiques. Les colonnes du temps d'exécution par Mvoxels (1 000 000 voxels) permettent d'avoir des temps qui peuvent être comparés d'une série à l'autre afin de vérifier la complexité des algorithmes.

L'algorithme d'augmentation du contraste est celui qui montre l'exécution la plus rapide de tous les traitements. Sa complexité d'ordre  $O(N)$ , où  $N$  est le nombre total de voxels, est appuyée par le temps constant de 0,214 secondes par Mvoxels pour toutes les séries. Le filtre de moyenne pondérée est le deuxième plus rapide algorithme. Son temps



d'exécution par Mvoxels se situe entre 1,952 et 2,232 secondes, ce qui donne une bonne piste de vérification pour la complexité de  $O(N K^3)$ , où  $K$  est le côté de la fenêtre du noyau, dans ce cas-ci égal à 3. Les variations de quelques dixièmes de secondes peuvent être attribuées aux éléments laissés de côté dans le calcul de l'ordre de grandeur de la complexité. Les algorithmes de dilatation et d'érosion, tous deux des opérations morphologiques fortement reliées, s'exécutent en un temps légèrement supérieur au filtre de moyenne pondérée. Leur temps par Mvoxels se situent entre 3,990 et 4,188 secondes, ce qui donne encore une fois une bonne piste de vérification pour la complexité de  $O(N K^3)$ . La différence de temps entre les opérations morphologiques et le filtre de moyenne pondérée peut résider dans les éléments ou constantes enlevés de l'ordre de grandeur de la complexité. Le filtre médian est le traitement qui s'exécute le moins rapidement car il a la complexité la plus importante ( $O(N K^3 \log K^3)$ ), due entre autres au tri des valeurs, et utilise une fenêtre de taille  $K = 5$ . Son temps d'exécution par Mvoxels est relativement constant (entre 8,533 et 8,756 secondes) d'une série à l'autre. L'algorithme de croissance de région diffère des autres car sa complexité est reliée non pas uniquement au nombre total de voxels mais également au nombre de voxels inclus dans la région. La proportion des voxels inclus dans la région par rapport au nombre total de voxels est très petite : environ 0,5 % pour la série de la crosse aortique et des carotides, 0,1 % pour la série de l'aorte et un peu moins de 1 % pour la série des données analytiques. Nous pouvons constater que le temps d'exécution de la croissance de région se situe entre l'augmentation du contraste et le filtre de moyenne pondérée. Cela semble bien confirmer que sa complexité se trouve entre  $O(N)$  et  $O(N K^3)$ .

Pour le moment et avec l'actuelle implantation des algorithmes, il peut être difficile d'utiliser les séquences en temps réel même si, malgré tout, les temps sont tout de même assez courts pour permettre une utilisation « agréable » (moins de trois minutes pour la série de données la plus volumineuse). Il est toutefois fortement envisageable d'optimiser les algorithmes afin, par exemple, d'éviter la plupart du temps de refaire certains calculs lorsqu'on passe d'un voxel à un autre dans les boucles des traitements.

De même, la parallélisation somme toute assez directe de ces opérations sur quelques processeurs permettrait de diminuer sensiblement le temps de d'exécution en le divisant presque exactement par le nombre de processeurs. Finalement, étant donné la puissance actuelle des processeurs graphiques modernes, on peut facilement envisager le transfert des traitements vers ces processeurs et espérer des gains significatifs en temps. De plus, dans ce dernier cas, les images générées par nos séquences de traitements seraient déjà chargées sur le processeur graphique et prêtes à l'affichage.

### **5.5 Discussion sur la méthodologie, l'hypothèse et les objectifs**

La méthodologie proposée pour reconstruire des surfaces 3D à partir de données médicales 3D, c'est-à-dire appliquer des séquences de filtres unitaires sur les données puis créer une isosurface, a permis de produire des résultats concrets. Dans certains cas, les résultats ont aussi été validés qualitativement et quantitativement. Les paragraphes suivants s'attardent sur quelques aspects particuliers de la méthodologie, puis revisitent l'hypothèse et les objectifs posés au premier chapitre.

Lors de la phase de conception du logiciel, nous avons défini une organisation quant à la répartition des responsabilités et la division en classes. La philosophie de cadre de travail logiciel et l'utilisation de patrons de conception de création et comportementaux ont poussé vers un partitionnement en un plus grand nombre de classes que pour d'autres approches (par exemple, une quinzaine de classes sont requises pour l'implantation des traitements présentés au chapitre 2). À l'exécution du programme, les objets doivent interagir fréquemment entre eux, ce qui affecte la rapidité du logiciel. Cette perte de rapidité se fait toutefois au profit d'une facilité attendue dans l'ajout de fonctionnalités et de support (comme un nouveau traitement ou un nouveau type de fichier de sortie) et d'une latitude dans le type de données traitées (3D, 4D, issues de simulations, etc.).

Nous avons implanté six traitements permettant d'améliorer la qualité des données sources et de les segmenter pour créer une isosurface modélisant l'organe désiré. La sélection des traitements a été basée sur leur simplicité et leur complémentarité en termes d'utilité. La composition des séquences de traitements pour les différentes séries de données sources, de même que le choix des paramètres, a été effectuée grandement par essais et erreurs. Le choix des essais a été fondé sur l'observation des caractéristiques des données sources et les effets des traitements en tant qu'unités individuelles. Nous avons présenté les séquences de traitements pour les trois séries de données, issues des essais, offrant des résultats prometteurs. Nous n'avons pas fait d'étude exhaustive sur la sensibilité des valeurs des paramètres de chaque traitement et les effets de l'ordre dans lequel les traitements sont agencés. Une telle étude serait pertinente pour évaluer la possibilité d'automatiser le processus de modélisation. Notre logiciel et ses diverses classes permettraient alors d'intégrer facilement cette automatisation du choix des traitements.

Afin d'évaluer quantitativement les résultats produits par le système, nous avons choisi deux métriques : l'aire et le volume interne des surfaces. Elles offrent des mesures de comparaison globales et, combinées aux comparaisons qualitatives des images traitées par rapports aux images originales et des surfaces par rapports aux géométries *in vivo* et aux résultats de segmentation manuelle, elles permettent d'émettre un jugement éclairé sur la performance du système à modéliser les vaisseaux sanguins. L'implantation du calcul de volume nécessite une surface fermée. Nous avons opté pour la solution simple d'offrir la possibilité d'ajouter des plans aux voxels d'intensité nulle au début et à la fin d'une série de données. De cette manière, l'algorithme de création de l'isosurface (méthode des cubes baladeurs) reste intouché, mais cela peut conduire à des surfaces ayant des extrémités légèrement plus petites à l'échelle d'un voxel. Lorsque la taille de l'objet modélisé est relativement petite, comme c'est le cas de l'aorte dans la série aorte qui ne comprend que cinq images sources, cette façon de fermer la surface est à éviter. Des métriques de validation plus locales, tels que l'aire de sections ou le degré de

courbure locale, compléteraient l'évaluation quantitative des résultats en mettant l'accent sur certains détails des objets modélisés.

Nous avons cherché à développer une approche uniformisante de la méthode de segmentation directe, qui travaille directement sur les données sources en trois dimensions, et des modèles déformables tel que le ballon qui, de par leurs contraintes intrinsèques, imposent un effet régulateur sur la surface produite. Notre approche consiste à travailler directement dans le format des données sources en 3D en conservant des valeurs décimales entre 0,0 et 1,0, tout en incorporant un mécanisme régulateur à travers l'un des traitements disponibles, l'algorithme de croissance de région. Ne pas binariser les voxels (leur imposer l'une des deux valeurs possibles de 0 ou 1) permet de créer directement une isosurface lisse sans la nécessité d'appliquer des algorithmes de lissage sur les polygones une fois la surface créée. Cela offre également un positionnement de la surface plus précis car il est possible de gérer la position de l'isosurface à l'échelle plus petite que le voxel. L'effet régulateur de l'algorithme de croissance de région tel qu'il est actuellement implanté s'exerce à très petite échelle, car le voisinage considéré pour la prise de décision quant à l'inclusion ou l'exclusion d'un voxel est de  $3 \times 3 \times 3$  voxels. Il est très réaliste de songer à considérer un voisinage plus grand sans que cela n'affecte trop dramatiquement le temps d'exécution de l'algorithme. C'est l'étude des différentes configurations possibles pour un voisinage de  $5 \times 5 \times 5$  voxels ou même  $9 \times 9 \times 9$  voxels afin de déterminer les règles d'inclusion et d'exclusion qui demandera beaucoup de temps à l'humain!

L'hypothèse de travail était la suivante : travailler directement sur les valeurs des données 3D, c'est-à-dire conserver le format « bain de voxels », constitue une approche uniformisante de la méthode de segmentation directe et des modèles déformables 3D tel que le ballon, permettant de profiter de la cohérence des données sources et de retrouver un modèle géométrique réaliste par des opérations simples. Les éléments des différents chapitres de ce mémoire permettent de confirmer cette hypothèse. Le chapitre 3 sur la

mise en oeuvre logicielle atteste que le premier objectif (développer un cadre de travail permettant de tester plusieurs algorithmes et facilitant l'ajout de nouvelles fonctionnalités ou méthodes alternatives et favoriser la réutilisation de design et de code pour produire un outil employable dans d'autres contextes d'application) a été atteint. Le chapitre 2 présentant les traitements sur les données sources atteste que le second objectif (implanter sous la forme de filtres des traitements unitaires applicables sur des bains de voxels) a été atteint. Le quatrième chapitre sur les résultats permet d'affirmer que le troisième objectif (proposer des séquences de traitements permettant de retrouver une surface tridimensionnelle pour quelques groupes de données sources types) a été atteint. Enfin, le chapitre 5 proposant une discussion du projet atteste que le dernier objectif (valider qualitativement et quantitativement le système de modélisation d'une surface à l'aide de modèles connus (analytiques) et réels) a également été atteint.

## CONCLUSION

La modélisation tridimensionnelle d'organes à partir d'informations physiques réelles provenant d'images médicales *in vivo* apporte des voies supplémentaires d'étude et d'analyse pour de nombreuses applications du domaine biomédical. La fidélité des modèles reconstruits est une condition essentielle et pourtant difficile à atteindre et à vérifier car le processus d'acquisition des images médicales introduit des artéfacts sur les données sources, et il n'existe pas de gabarit *in vivo* des organes permettant une validation quantitative des modèles.

Ce mémoire se situe au coeur de cette problématique et propose une méthodologie de reconstruction de modèles tridimensionnels à partir de données tridimensionnelles à travers l'application spécifique de création de surfaces 3D de vaisseaux sanguins à partir données produites par IRM. Il a proposé une piste quelque peu différente de la majorité des méthodes présentées dans la revue bibliographique : améliorer la fidélité du modèle par le traitement des données sources dans leur format initial et créer une surface polygonale finale du modèle en dernière étape du processus. L'hypothèse de travail posée au premier chapitre était la suivante :

- Travailler directement sur les valeurs des données 3D, c'est-à-dire conserver le format « bain de voxels », constitue une approche uniformisante de la méthode de segmentation directe et des modèles déformables 3D tel que le ballon, permettant de profiter de la cohérence des données sources et de retrouver un modèle géométrique réaliste par des opérations simples.

Quatre objectifs spécifiques ont été définis pour permettre de vérifier cette hypothèse :

- 1) développer un cadre de travail permettant de tester plusieurs algorithmes et facilitant l'ajout de nouvelles fonctionnalités ou méthodes alternatives, et favoriser la réutilisation de design et de code pour produire un outil employable dans d'autres contextes d'application;
- 2) implanter sous la forme de filtres des traitements unitaires applicables

sur des bains de voxels; 3) proposer des séquences de traitements permettant de retrouver une surface tridimensionnelle pour quelques groupes de données sources types; 4) valider qualitativement et quantitativement le système de modélisation d'une surface à l'aide de modèles connus (analytiques) et réels. Ces objectifs ont été atteints de façon progressive à travers les différents chapitres de ce mémoire.

Dans le chapitre 2, nous avons introduit plusieurs traitements unitaires simples, applicables sur des données sources tridimensionnelles : l'augmentation du contraste, le filtre de moyenne pondérée, le filtre médian, les opérations morphologiques (dilatation et érosion) et la croissance de région. Ces traitements, qui améliorent et segmentent les données sources en tirant profit de la cohérence spatiale 3D des données, constituent une approche uniformisante entre la méthode de segmentation directe et les modèles déformables tel que le ballon. En effet, ces traitements travaillent directement dans le format des données sources en 3D comme la méthode de segmentation directe, en conservant toutefois des valeurs décimales entre 0,0 et 1,0 pour avoir une plus grande latitude d'opération, tout en incorporant un mécanisme régulateur semblable à celui du ballon à travers l'un des traitements disponibles, l'algorithme de croissance de région.

Nous avons exposé la mise en oeuvre logicielle du projet au chapitre 3. L'architecture logicielle, de par l'organisation et la responsabilité de ses classes, facilite l'ajout de nouvelles méthodes et de nouvelles fonctionnalités tout en offrant le cadre de travail pour développer des applications semblables à la modélisation 3D de vaisseaux sanguins à partir d'images médicales, comme par exemple la modélisation de nuages 3D de concentrations de gaz dans une pièce à partir de données de simulations numériques.

Dans le quatrième chapitre, nous avons présenté les résultats produits par notre système pour l'application spécifique de la modélisation 3D de vaisseaux sanguins à partir d'images médicales à l'aide de trois séries de données différentes : IRM de la crosse aortique et des carotides, IRM de l'aorte et données analytiques de la crosse aortique et

de l'aorte. Les surfaces 3D résultantes ont été créées après l'exécution rapide des séquences de traitements, les traitements ayant été choisis parmi ceux introduits au chapitre 2.

Le chapitre 5 a été consacré à la discussion de la méthodologie et à l'évaluation des résultats de manière qualitative et quantitative. La validation qualitative s'est basée sur des comparaisons entre les images segmentées par notre logiciel et les images médicales sources, des comparaisons entre les surfaces produites et les géométries *in vivo*, et des comparaisons entre les surfaces produites et les surfaces issues d'une segmentation manuelle. Nous pouvons affirmer que les surfaces 3D produites par notre logiciel sont d'allure lisse et généralement en accord avec l'anatomie humaine. Les séquences de traitements ont généralement été efficaces pour éliminer le bruit présent sur les données sources et pour produire des objets d'intensité uniforme et aux frontières relativement lisses. Les surfaces résultantes présentent parfois des petites régions incorrectement fusionnées et des extrémités trop arrondies, mais les tests ont montré qu'il est possible de limiter, voire éliminer ces problèmes par les choix appropriés des paramètres des traitements. La validation quantitative s'est basée sur deux métriques : l'aire et le volume interne des surfaces. Nous avons été en mesure d'appliquer cette validation sur deux des trois séries de données en calculant le pourcentage d'erreur des surfaces produites par notre logiciel par rapport à la surface reconstruite à partir d'une segmentation manuelle et à la surface réelle du modèle mathématique ayant donné lieu aux données analytiques. La validation quantitative n'a pas été concluante dans le premier cas et, dans le cas des données analytiques, a montré des erreurs minimales.

La méthodologie proposée, c'est-à-dire appliquer des séquences de filtres unitaires sur les données sources puis créer une isosurface pour obtenir le modèle recherché, a permis de produire des résultats concrets et acceptables. L'hypothèse a été vérifiée.



Au fil du mémoire, nous avons mentionné plusieurs avenues intéressantes d'améliorations et de complétion du projet. Parmi celles-ci, retenons l'implantation des séquences de traitements sur des processeurs graphiques modernes, qui offrirait certainement des gains importants en temps d'exécution et ouvrirait le chemin vers l'utilisation du logiciel en temps réel. Également, il serait fort pertinent de développer l'implantation de nouveaux algorithmes de traitements (tels que le lissage par diffusion anisotrope ou des versions adaptatives des traitements existants agissant différemment selon le rayon de courbure local des objets) et l'implantation des versions 4D des différents sous-systèmes afin de pouvoir modéliser des données qui évoluent dans le temps. Enfin, une étude de sensibilité des paramètres des traitements fournirait les outils de base vers une voie d'automatisation du processus de modélisation au sein de notre logiciel.

## BIBLIOGRAPHIE

- AZUMA, Daniel. 2000. The GLOW toolkit. In *Site du SourceForge.net*. [En ligne].  
<http://glow.sourceforge.net/>  
 (Page consultée le 1<sup>er</sup> mars 2006)
- BLOCH, Isabelle. 2001. « Morphologie mathématique ». *Le traitement des images*. Sous la direction de Henri Maître. Paris : Hermes Science. 111–153.
- BOSE, Tamal. 2004. *Digital signal and image processing*. Hoboken, New Jersey : Wiley. 706p.
- CASELLES, Vicent, CATTÉ, Francine, COLL, Tomeu, DIBOS, Françoise. 1993. « A geometric model for active contours in image processing ». *Numerische Mathematik*. 66 : 1. 1–31.
- CASELLES, Vicent, KIMMEL, Ron, SAPIRO, Guillermo. 1997. « Geodesic active contours ». *International Journal of Computer Vision*. 22 : 1. 61–79.
- CEBRAL, Juan Raul, LÖHNER, Rainald. 2001. « From medical images to anatomically accurate finite element grids ». *International Journal for Numerical Methods in Engineering*. 51 : 8. 985–1008.
- CHEN, Yang, MEDIONI, Gérard. 1995. « Description of complex objects from multiple range images using an inflating balloon model ». *Computer Vision and Image Understanding*. 61 : 3. 325–334.
- COATRIEUX, Jean-Louis, ROUX, Christian. 1997. « Biomedical imaging : integration of image engineering, biology and medicine ». *Contemporary perspectives in three-dimensional biomedical imaging*. Sous la direction de Christian Roux et Jean-Louis Coatrieux. Amsterdam : IOS Press. 3–27.

- COCQUEREZ, Jean-Pierre, PHILIPP, Sylvie. 1995. *Analyse d'images : filtrage et segmentation*. Paris : Masson. 457p.
- COHEN, Laurent D., COHEN, Isaac. 1990. « A finite element method applied to new active contour models and 3D reconstructions from cross sections ». *Proceedings of the Third International Conference on Computer Vision*. Osaka : IEEE. 587–591.
- COHEN, Laurent D., COHEN, Isaac. 1993. « Finite element methods for active contour models and balloons for 2D and 3D images ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 15 : 11. 1131–1147.
- CORMEN, Thomas H., LEISERSON, Charles E., RIVEST, Ronald L., STEIN, Clifford. 2001. *Introduction to algorithms*. 2<sup>e</sup> édition. Montréal : McGraw-Hill. 1180p.
- DEITEL, Harvey M., DEITEL, Paul J. 1998. *C++ how to programm*. 2<sup>e</sup> édition. Upper Saddle River, New Jersey : Prentice Hall. 1130p.
- DEMIRIS, Athanasios, MAYER, Achim, MEINZER, Hans Peter. 1997. « 3-D visualization in medicine : an overview ». *Contemporary perspectives in three-dimensional biomedical imaging*. Sous la direction de Christian Roux et Jean-Louis Coatrieux. Amsterdam : IOS Press. 79–105.
- DESBRUN, Mathieu, MEYER, Mark, SCHRÖDER, Peter, BARR, Alan H. 1999. « Implicit fairing of irregular meshes using diffusion and curvature flow ». *Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99)*. Los Angeles, California : ACM. 317–324.
- DUNCAN, James S., AYACHE, Nicholas. 2000. « Medical image analysis : progress over two decades and the challenges ahead ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22 : 1. 85–106.

- FISHER, Robert, PERKINS, Simon, WALKER, Ashley, WOLFART, Erik. 2004. Point Operations - Contrast Stretching. In *Site du Hypermedia Image Processing Reference 2*. [En ligne]. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm#1> (Page consultée le 6 décembre 2005)
- FLEISHMAN, Shachar, DRORI, Iddo, COHEN-OR, Daniel. 2003. « Bilateral mesh denoising ». *ACM Transactions on Graphics*. 22 : 3. 950–953.
- FRANGI, Alejandro F., NIESSEN, Wiro J., HOOGEVEEN, Romhild M., VAN WALSUM, Theo, VIERGEVER, Max A. 1999. « Model-based quantitation of 3-D magnetic resonance angiographic images ». *IEEE Transactions on Medical Imaging*. 18 : 10. 946–956.
- FREY, Pascal Jean, GEORGE, Paul-Louis. 1999. *Maillages : applications aux éléments finis*. Paris : Hermes. 840p.
- FREY, Pascal Jean. 2001. *YAMS : A fully automatic adaptive isotropic surface remeshing procedure*. Rocquencourt, France : Institut National de Recherche en Informatique et en Automatique. 36p. 0252.
- GAMMA, Erich, HELM, Richard, JOHNSON, Ralph, VLISSIDES, John. 1995. *Design patterns : elements of reusable object-oriented software*. Don Mills, Ontario : Addison-Wesley. 395p.
- GEORGE, Louis-Paul, BOROUCHAKI, Houman. 1997. *Triangulation de Delaunay et maillage : applications aux éléments finis*. Paris : Hermes. 432p.
- GERIG, Guido, KÜBLER, Olaf, KIKINIS, Ron, JOLESZ, Ferenc A. 1992. « Nonlinear anisotropic filtering of MRI data ». *IEEE Transactions on Medical Imaging*. 11 : 2. 221–232.

- GILL, Jeremy D., LADAK, Hanif M., STEINMAN, David A., FENSTER, Aaron. 2000. « Accuracy and variability assessment of a semiautomatic technique for segmentation of the carotid arteries from three-dimensional ultrasound images ». *Medical Physics*. 27 : 6. 1333–1342.
- GOURAUD, Henri. 1971. « Continuous shading of curved surfaces ». *IEEE Transactions on Computers*. 20 : 6. 623–628.
- HERMENT, A., PELLOT, C., GIOVANNELLI, J.F. 1997. « Application de méthodes régularisantes en imagerie cardio-vasculaire ». *Traitement d'images médicales : du pixel à la structure*. Édité par Yves Goussard. Montréal : Éditions de l'École Polytechnique de Montréal. 29–58.
- INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE. 2004. Project-Team-Gamma (Logiciel YAMS). In *Site de l'Institut National de Recherche en Informatique et en Automatique*. [En ligne]. <http://www.inria.fr/rapportsactivite/RA2004/gamma2004/uid32.html> (Page consultée le 1<sup>er</sup> mars 2006)
- INVISU INC. 2001. VU, logiciel de visualisation scientifique. In *Site de inVisu inc.* [En ligne]. <http://www.invisu.ca/f/vu.php> (Page consultée le 1<sup>er</sup> mars 2006)
- JIANG, Tianzi, FAN, Yong. 2002. « Parallel genetic algorithm for 3D medical image analysis ». *IEEE International Conference on Systems, Man and Cybernetics*. Yasmine Hammamet, Tunisie : IEEE. 6.
- KASS, Michael, WITKIN, Andrew, TERZOPOULOS, Demetri. 1987. « Snakes : Active contour models ». *International Journal of Computer Vision*. 1 : 4. 321–331.

- KENNEDY, David N., FILIPEK, Pauline A., CAVINESS JR., Verne S. 1989. « Anatomic segmentation and volumetric calculations in nuclear magnetic resonance imaging ». *IEEE Transactions on Medical Imaging*. 8 : 1. 1–7.
- KIRBAS, Cemil, QUEK, Francis K. H. 2003. « Vessel extraction in medical images by 3D wave propagation and traceback ». *Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)*. Bethesda, Maryland : IEEE. 174–181.
- KITWARE. 2006. VTK home page. In *Site de Kitware inc.* [En ligne]. <http://public.kitware.com/VTK/index.php>  
(Page consultée le 2 mars 2006)
- LADAK, Hanif M., MILNER, Jaques S., STEINMAN, David A. 2000. « Rapid 3D segmentation of the carotid bifurcation from serial MR images ». *Journal of Biomechanical Engineering*. 122 : 1. 96–99.
- LARMAN, Craig. 2002. *Applying UML and patterns : an introduction to object-oriented analysis and design and the unified process*. 2<sup>e</sup> édition. Upper Saddle River, New Jersey : Prentice Hall PTR. 627p.
- LEVENTON, Michael E., FAUGERAS, Olivier, GRIMSON, W. Eric L., WELLS III, William M. 2000. « Level set based segmentation with intensity and curvature priors ». *Proceeding of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA)*. Hilton Head Island, South Carolina : IEEE. 4–11.
- LINCOURT-ÉTHIER, Marie-Noëlle. 2005. *Approximation du système vasculaire : évaluation de l'erreur de modélisation*. 120p. Mémoire de maîtrise en génie biomédical, École Polytechnique de Montréal.

- LIU, Xinguo, BAO, Hujun, HENG, PhengAnn, WONG, TienTsin, PENG, Qunsheng. 2001. « Constrained fairing for meshes ». *Computer Graphics Forum*. 20 : 2. 115–123.
- LIU, Xinguo, BAO, Hujun, SHUM, Heung-Yeung; PENG, Qunsheng. 2002. « A novel volume constrained smoothing method for meshes ». *Graphical Models*. 64 : 3-4. 169–182.
- LORENSEN, William E., CLINE, Harvey E. 1987. « Marching cubes: A high-resolution 3D surface construction algorithm ». *ACM Computer Graphics*. 21 : 4. 163–170.
- MAÎTRE, Henri. 2003. « Prétraitements ». *Le traitement des images*. Sous la direction de Henri Maître. Paris : Hermes Science. 239–257.
- MAÎTRE, Henri. 2003. « Segmentation par régions ». *Le traitement des images*. Sous la direction de Henri Maître. Paris : Hermes Science. 287–307.
- MALLADI, Ravikanth, SETHIAN, James A., VEMURI, Baba C. 1995. « Shape modeling with front propagation : A level set approach ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 17 : 2. 158–175.
- MALLADI, Ravikanth, KIMMEL, Ron, ADALSTEINSSON, David, SAPIRO, Guillermo, CASELLES, Vicent, SETHIAN, James A. 1996. « A geometric approach to segmentation and analysis of 3D medical images ». *Proceeding of the Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'96)*. San Fransisco : IEEE. 244–252.
- MASHIKO, Takashi, YAGOU, Hirokazu, WEI, Darning, DING, Youdong, WU, Genfeng. 2004. « 3D triangle mesh smoothing via adaptive MMSE filtering ». *Proceedings of the Fourth International Conference on Computer and Information Technology (CIT 2004)*. Wuhan, Chine : IEEE. 734–740.

- MATSUSHITA, Kenji, KANEKO, Toyohisa. 2000. « Surface reconstruction with preserving volume data ». *Systems and Computers in Japan*. 31 : 12. 43–51.
- MCINERNEY, Tim, TERZOPOULOS, Demetri. 1996. « Deformable models in medical image analysis : a survey ». *Medical Image Analysis*. 1 : 2. 91–108.
- MEYER-BÄSE, Anke. 2004. *Pattern recognition for medical imaging*. Amsterdam : Elsevier Academic Press. 386p.
- MOORE, J.A., STEINMAN, D.A., HOLDSWORTH, D.W., ETHIER, C.R. 1999. « Accuracy of computational hemodynamics in complex arterial geometries reconstructed from magnetic resonance imaging ». *Annals of Biomedical Engineering*. 27. 32–41.
- NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. 2006. DICOM Homepage. In *Site du Digital Imaging and Communications in Medecine*. [En ligne]. <http://dicom.nema.org/>  
(Page consultée le 1<sup>er</sup> mars 2006)
- NIESSEN, Wiro J., ROMENY, Bart M. H., VIERGEVER, Max A. 1998. « Geodesic deformable models for medical image analysis ». *IEEE Transactions on Medical Imaging*. 17 : 4. 634–641.
- NIKOLAIDIS, Nikos, PITAS, Ioannis. 2001. *3-D image processing algorithms*. Toronto : Wiley-Interscience. 177p.
- OHTAKE, Yutaka, BELYAEV, Alexander, SEIDEL, Hans-Peter. 2002. « Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals ». *Vision, Modeling and Visualization (VMV 2002)*. Erlangen, Allemagne : IOS Press. 203–210.



- OPENGL. 2004. GLUT - The OpenGL utility toolkit. In *Site de OpenGL.org*. [En ligne].  
<http://www.opengl.org/resources/libraries/glut.html>  
 (Page consultée le 1<sup>er</sup> mars 2006)
- PARAGIOS, Nikos, DERICHE, Rachid. 1999. « Unifying boundary and region-based information for geodesic active tracking ». *Computer Society Conference on Computer Vision and Pattern Recognition*. Colorado : IEEE. 2, 300–305.
- PARDO, Xose Manuel, RADEVA, Petia, CABELLO, Diego. 2003. « Discriminant snakes for 3D reconstruction of anatomical organs ». *Medical Image Analysis*. 7 : 3. 293–310.
- PRATT, William K. 2001. *Digital image processing : PIKS inside*. 3<sup>e</sup> édition. New York : Wiley-Interscience. 735p.
- PRESSMAN, Roger S. 2001. *Software engineering : a practitioner's approach*. 5<sup>e</sup> édition. Toronto : McGraw-Hill. 860p.
- RORDEN, Chris. 2002. ezDICOM free medical viewer software. In *Site du ezDICOM software*. [En ligne]. <http://www.sph.sc.edu/cmd/rorden/ezdicom.html>  
 (Page consultée le 1<sup>er</sup> mars 2006)
- RÖSSL, Christian, ZEILFELDER, Frank, NÜRNBERGER Günther, SEIDEL, Hans-Peter. 2004. « Reconstruction of volume data with quadratic super splines ». *IEEE Transactions on Visualization and Computer Graphics*. 10 : 4. 397–409.
- RUSS, John C. 1999. *The image processing handbook*. 3<sup>e</sup> édition. Boca Raton, Floride : CRC Press. 771p.
- SETHIAN, James A. 1999. *Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science*. Cambridge : Cambridge University Press. 378p.

- SILICON GRAPHICS, INC. 2005. SGI - services & support : standard template library programmer's guide. In *Site de Silicon Graphics, Inc.* [En ligne]. <http://www.sgi.com/tech/stl/>  
(Page consultée le 3 mars 2006)
- SHAHIDI, Ramin, TOMBROPOULOS, Rhea, GRZESZCZUK, Robert P. 1998. « Clinical applications of three-dimensional rendering of medical data sets ». *Proceedings of the IEEE*. 86 : 3. 555–568.
- SHEN, Yuzhong, BARNER, Kenneth E. 2004. « Fuzzy vector median-based surface smoothing ». *IEEE Transactions on Visualization and Computer Graphics*. 10 : 3. 252–265.
- STAIB, Lawrence H., DUNCAN, James S. 1996. « Model-based deformable surface finding for medical images ». *IEEE Transactions on Medical Imaging*. 15 : 5. 720–731.
- SUTTON, Bradley P., Douglas C., Jeffrey A. 2003. « Fast, iterative image reconstruction for MRI in the presence of field inhomogeneities ». *IEEE Transactions on Medical Imaging*. 22 : 2. 178–188.
- SZELISKI, Richard, TONNESEN, David, TERZOPOULOS, Demetri. 1993. « Modeling surfaces of arbitrary topology with dynamic particles ». *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*. Los Alamitos, California : IEEE. 82–87.
- TASDIZEN, Tolga, WHITAKER, Ross, BURCHARD, Paul, OSHER, Stanley. 2002. « Geometric surface smoothing via anisotropic diffusion of normals ». *Proceedings of Visualization 2002*. Boston, Massachusetts : IEEE. 125–132.

- TASDIZEN, Tolga, WHITAKER, Ross, BURCHARD, Paul, OSHER, Stanley. 2003. « Geometric surface processing via normal maps ». *ACM Transactions on Graphics*. 22 : 4. 1012–1033.
- TAUBIN, Gabriel. 1995. « A signal processing approach to fair surface design ». *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (SIGGRAPH'95)*. Los Angeles, California : ACM. 351–358.
- TAUBIN, Gabriel. 1995. « Curve and surface smoothing without shrinkage ». *Proceedings of the Fifth International Conference on Computer Vision*. MIT, Cambridge, Massachusetts : IEEE. 852–857.
- TAUBIN, Gabriel, ZHANG, Tong, GOLUB, Gene. 1996. « Optimal surface smoothing as filter design ». *Proceedings of the Fourth European Conference on Computer Vision (ECCV'96)*. Cambridge, Royaume-Uni : Springer-Verlag. 283–292.
- TAUBIN, Gabriel. 2001. « Linear anisotropic mesh filtering ». Yorktown Heights, États-Unis : IBM T.J. Watson Research Center. 12p. IBM Research Report RC22213.
- TEK, Hüseyin, KIMIA, Benjamin B. 1997. « Volumetric segmentation of medical images by three-dimensional bubbles ». *Computer Vision and Image Understanding*. 65 : 2. 246–258.
- VOLLMER, Jörg, MENCL, Robert, MÜLLER, Heinrich. 1999. « Improved Laplacian smoothing of noisy surface meshes », *Computer Graphics Forum*. 18 : 3. 131–138.
- WANG, K. C., TAYLOR, C. A., HSIAU, Z., PARKER, D., DUTTON, R. W. 1998. « Level set methods and MR image segmentation for geometric modeling in computational hemodynamics ». *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Hong Kong : IEEE. 20 : 6. 3079–3082.

- WANG, Kenneth, DUTTON, Robert W., TAYLOR, Charles A. 1999. « Improving geometric model construction for blood flow modeling ». *IEEE Engineering in Medicine and Biology Magazine*. 18 : 6. 33–39.
- WATT, Alan, POLICARPO, Fabio. 1998. *The computer image*. Don Mills, Ontario : Addison-Wesley. 751p.
- WEI, Min, ZHOU, Yongjin, WAN, Mingxi. 2004. « A fast snake model based on non-linear diffusion for medical image segmentation ». *Computerized Medical Imaging and Graphics*. 28 : 3. 109–117.
- WENG, Nan, YANG, Yee-Hong, PIERSON, Roger. 1997. « Three-dimensional surface reconstruction using optical flow for medical imaging ». *IEEE Transactions on Medical Imaging*. 16 : 5. 630–641.
- WOO, Mason, NEIDER, Jackie, DAVIS, Tom, SHREINER, Dave. 2000. *OpenGL*. 3<sup>e</sup> édition. Paris : CampusPress France. 639p.
- XU, Chenyang, PRINCE, Jerry L. 1998. « Snakes, shapes, and gradient vector flow ». *IEEE Transactions on Image Processing*. 7 : 3. 359–369.
- YAGOU, Hirokazu, OHTAKE, Yutaka, BELYAEV, Alexander G. 2002. « Mesh smoothing via mean and median filtering applied to face normals ». *Proceedings of Geometric Modeling and Processing 2002*. Saitama, Japon : IEEE. 124–131.
- YIM, Peter J., CEBRAL, Juan J., MULLICK, Rakesh, MARCOS, Hani B., CHOYKE, Peter L. 2001. « Vessel surface reconstruction with a tubular deformable model ». *IEEE Transactions on Medical Imaging*. 20 : 12. 1411–1421.

## ANNEXE A : DONNÉES SOURCES MÉDICALES

Cette annexe présente les images médicales des deux séries utilisées comme données sources dans ce projet. La première série comporte 140 images de la crosse aortique et des carotides et la seconde 5 images de l'aorte. L'ordre des images est de gauche à droite puis du haut vers le bas. La résolution originale des images est de 256 x 256 pixels.

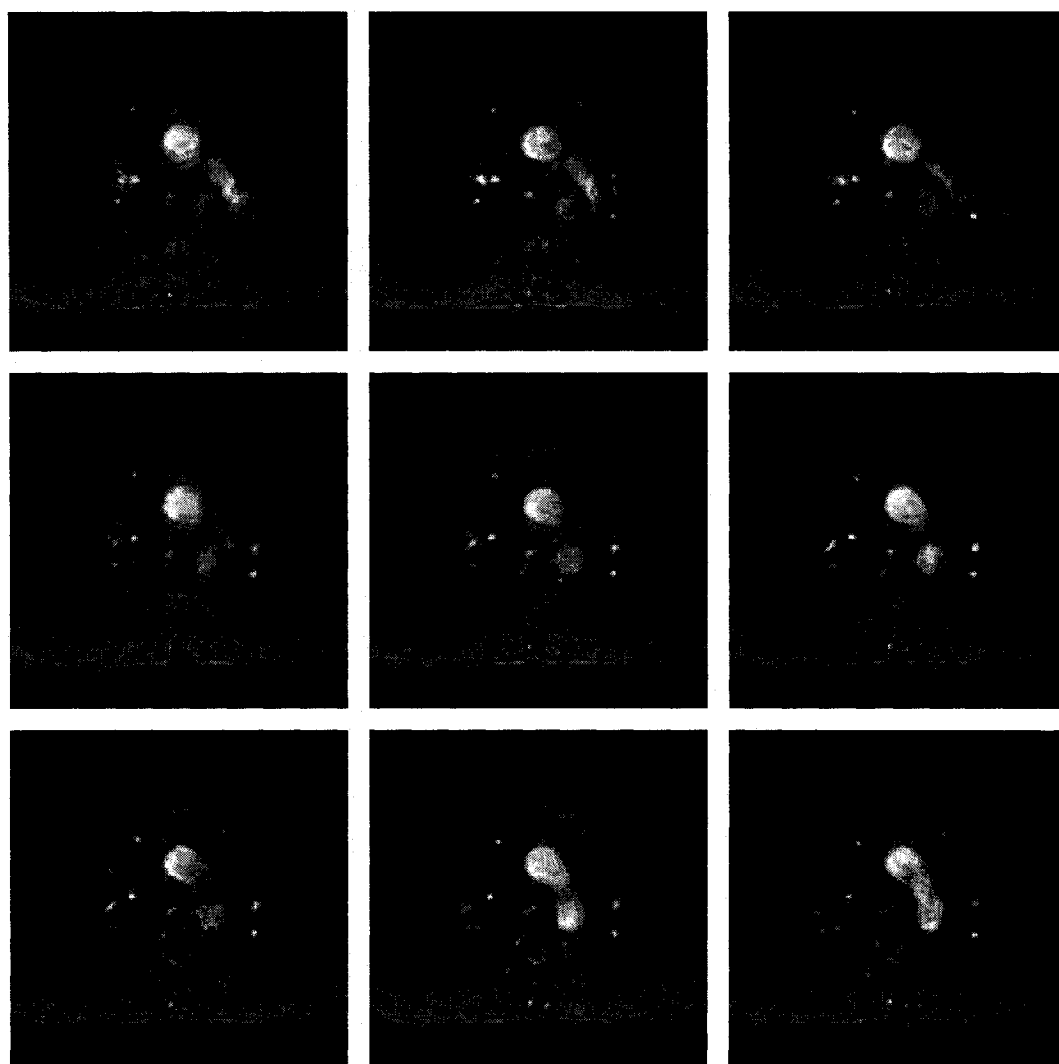
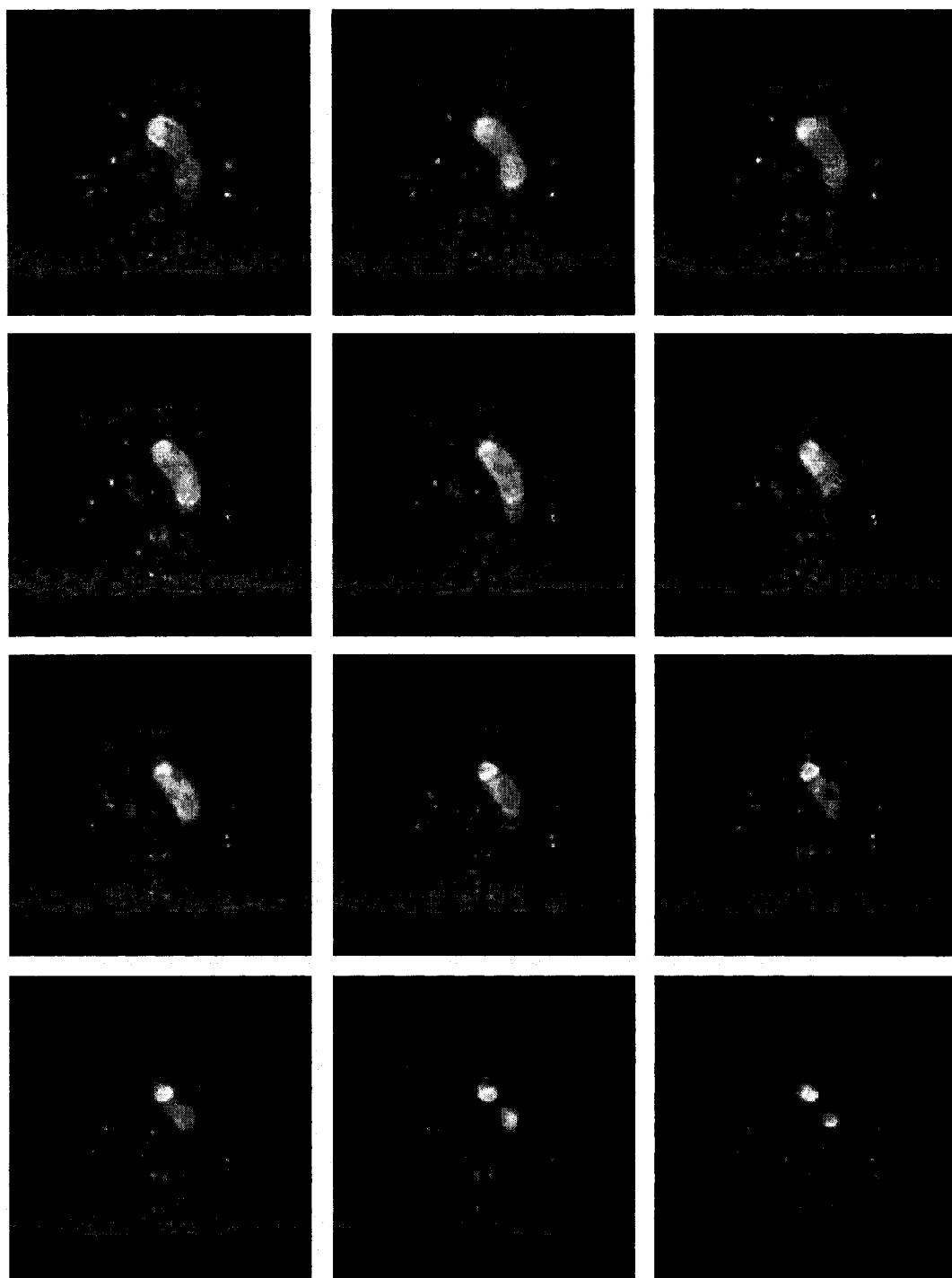


Figure A.1 : Images médicales de la série crosse aortique et carotides. Images 1 à 9.



**Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 10 à 21.**

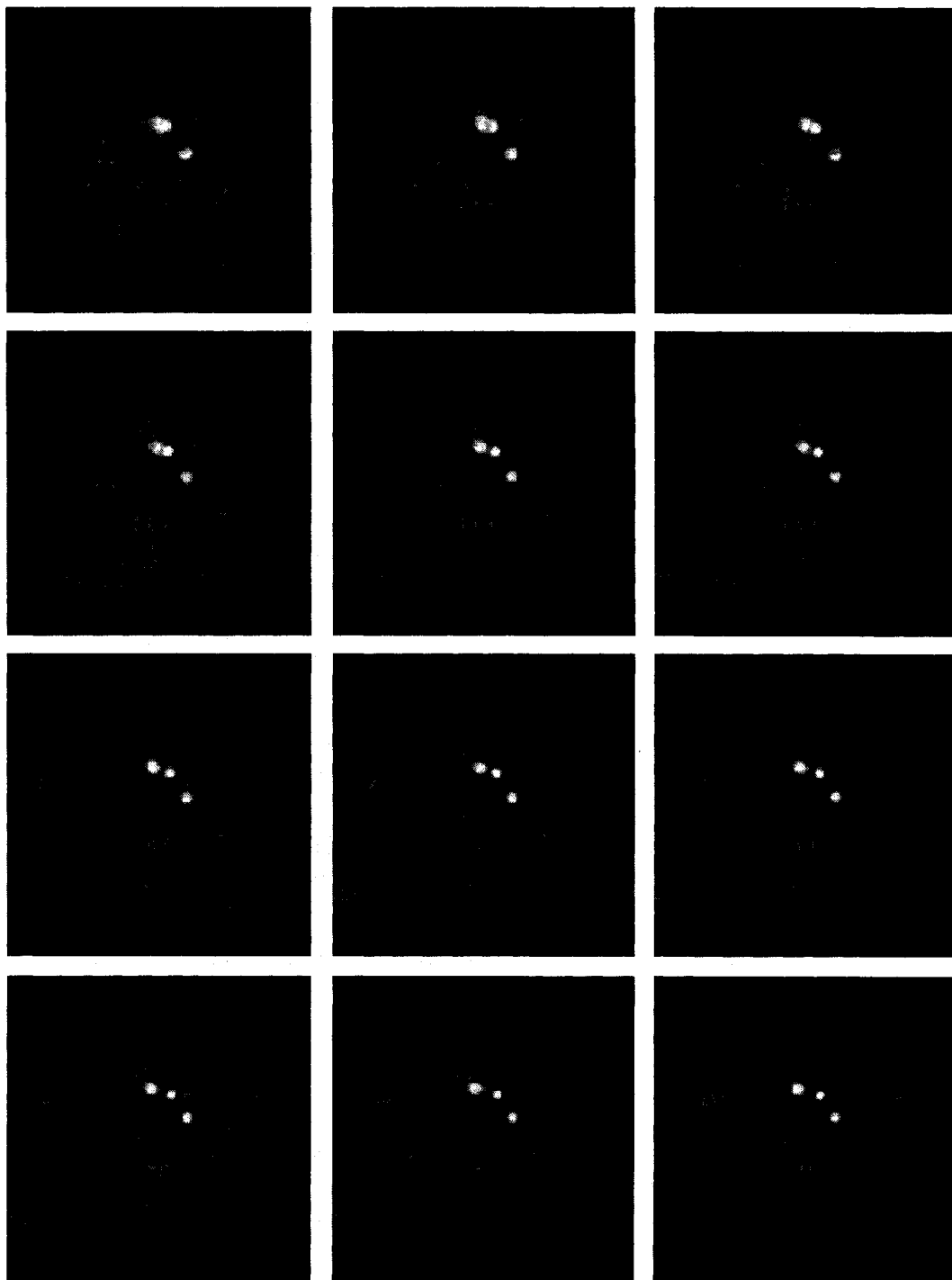


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 22 à 33.

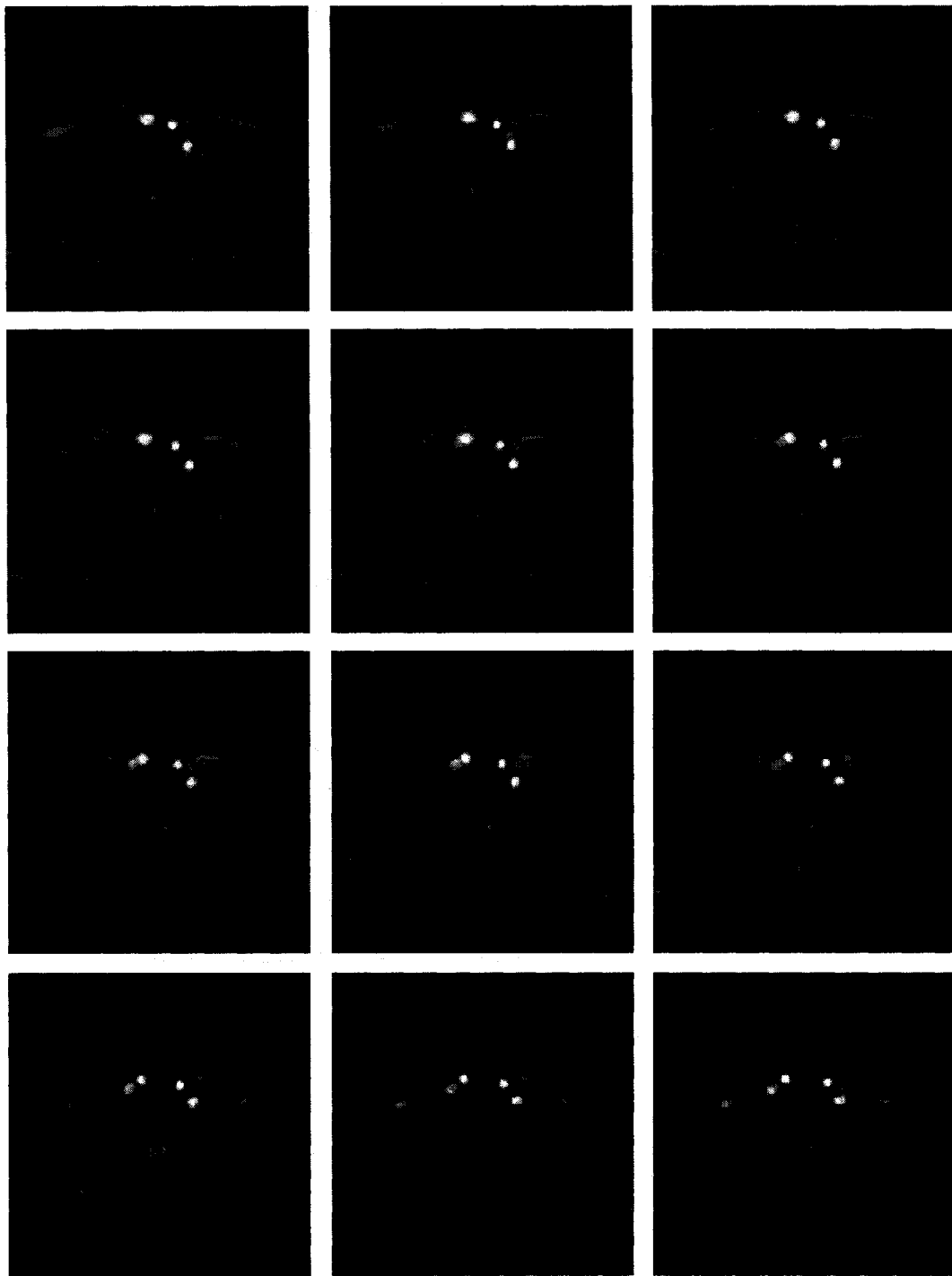


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 34 à 45.



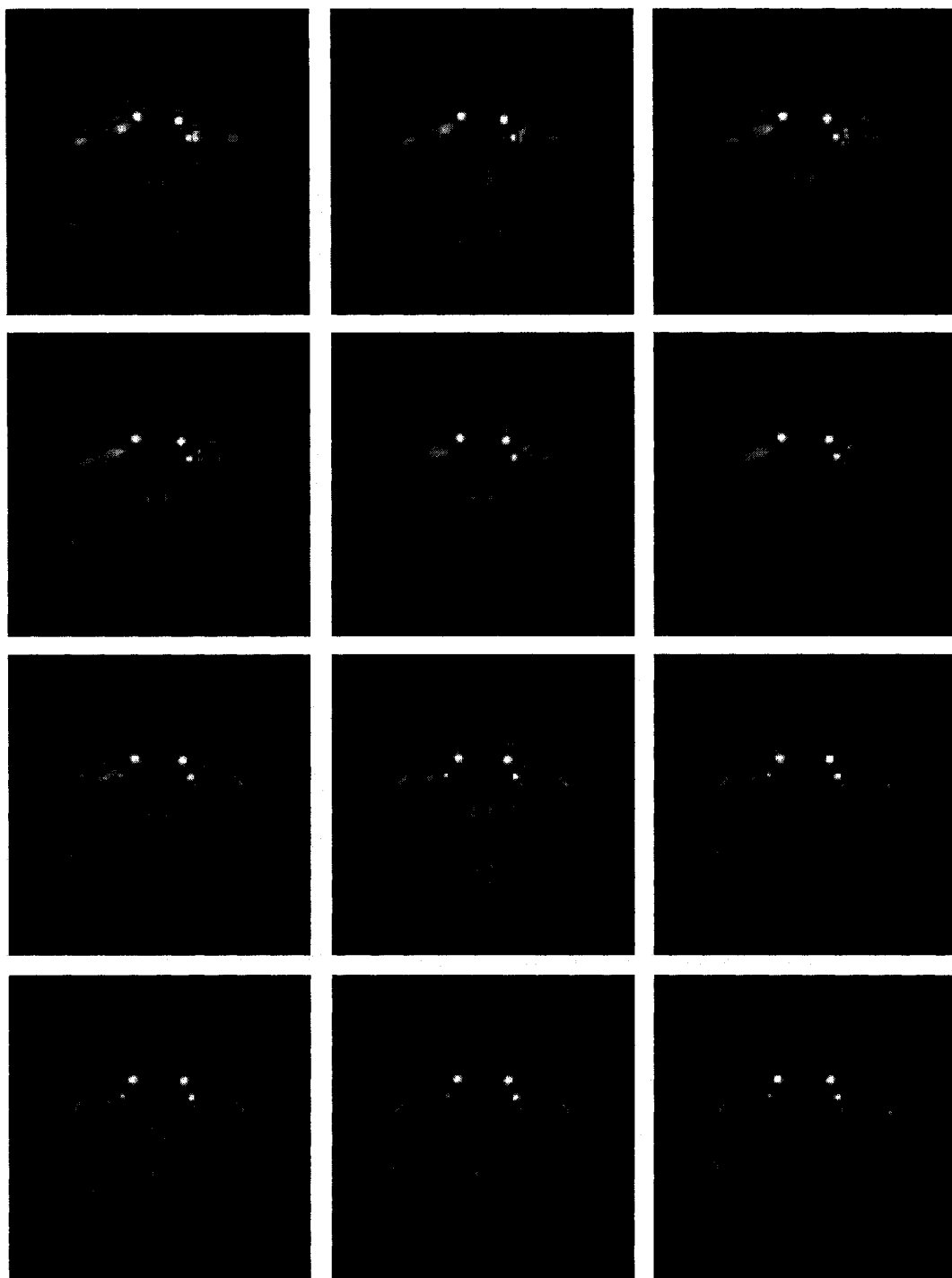


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 46 à 57.

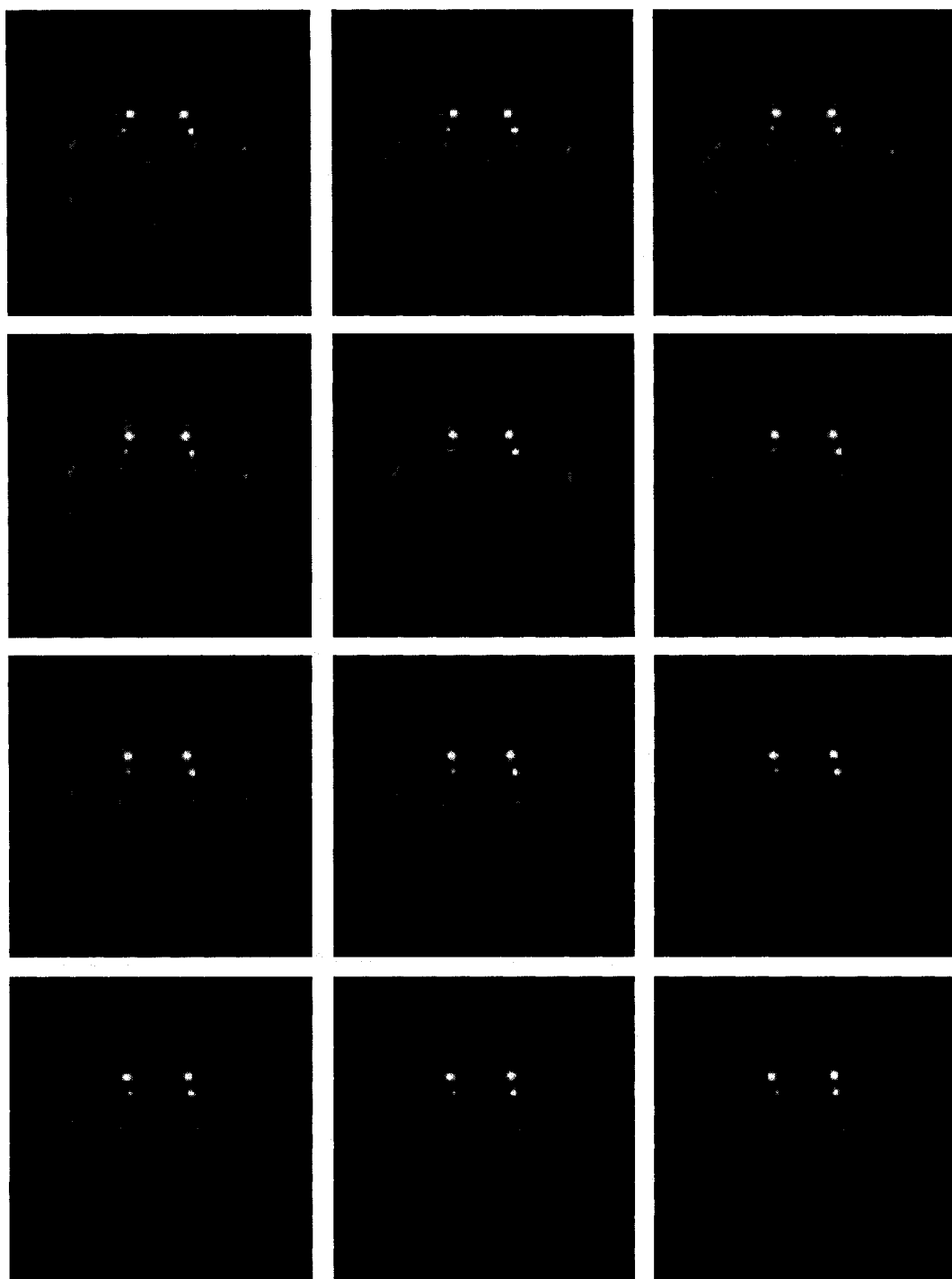


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 58 à 69.

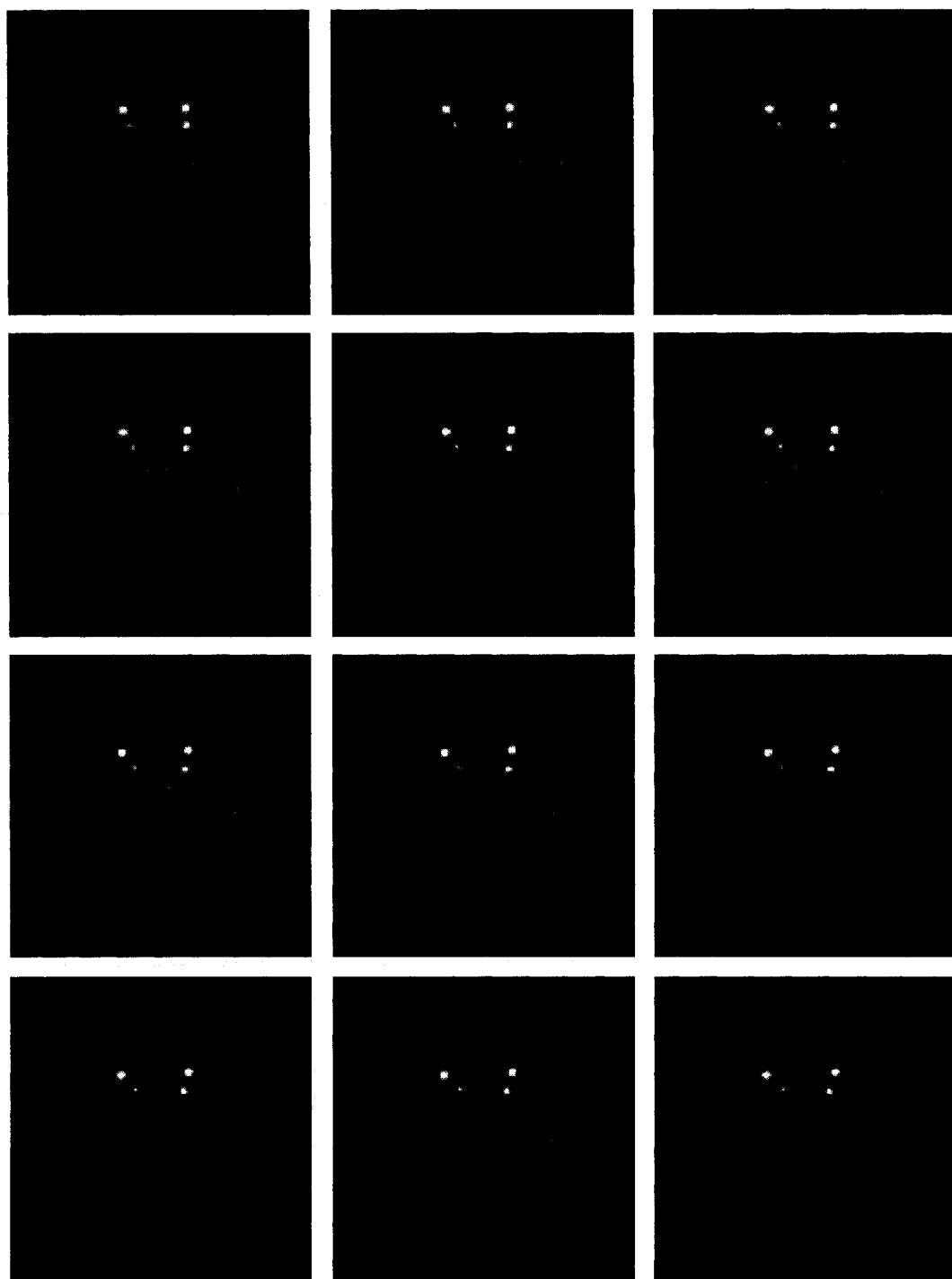


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 70 à 81.

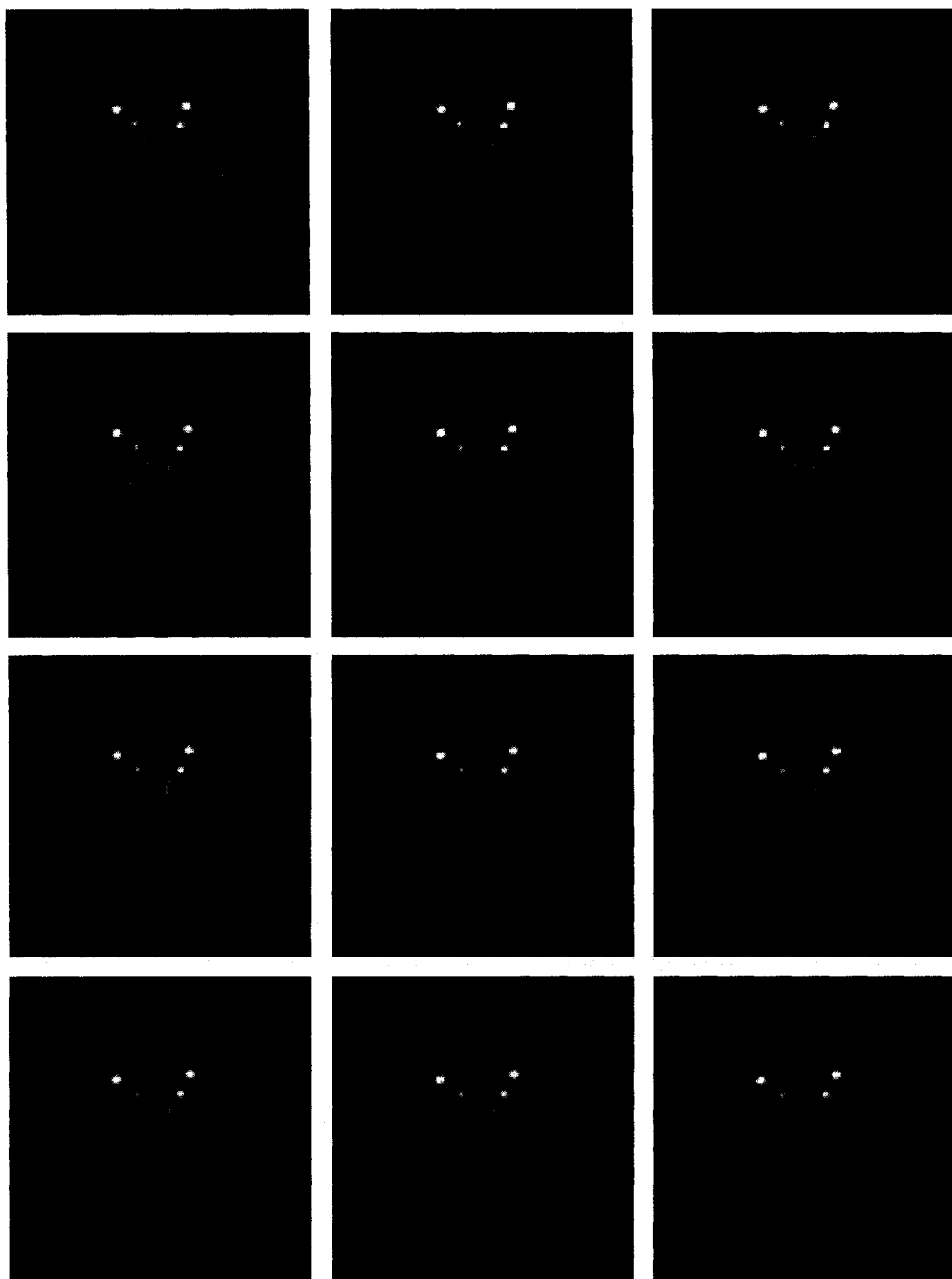


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 82 à 93.

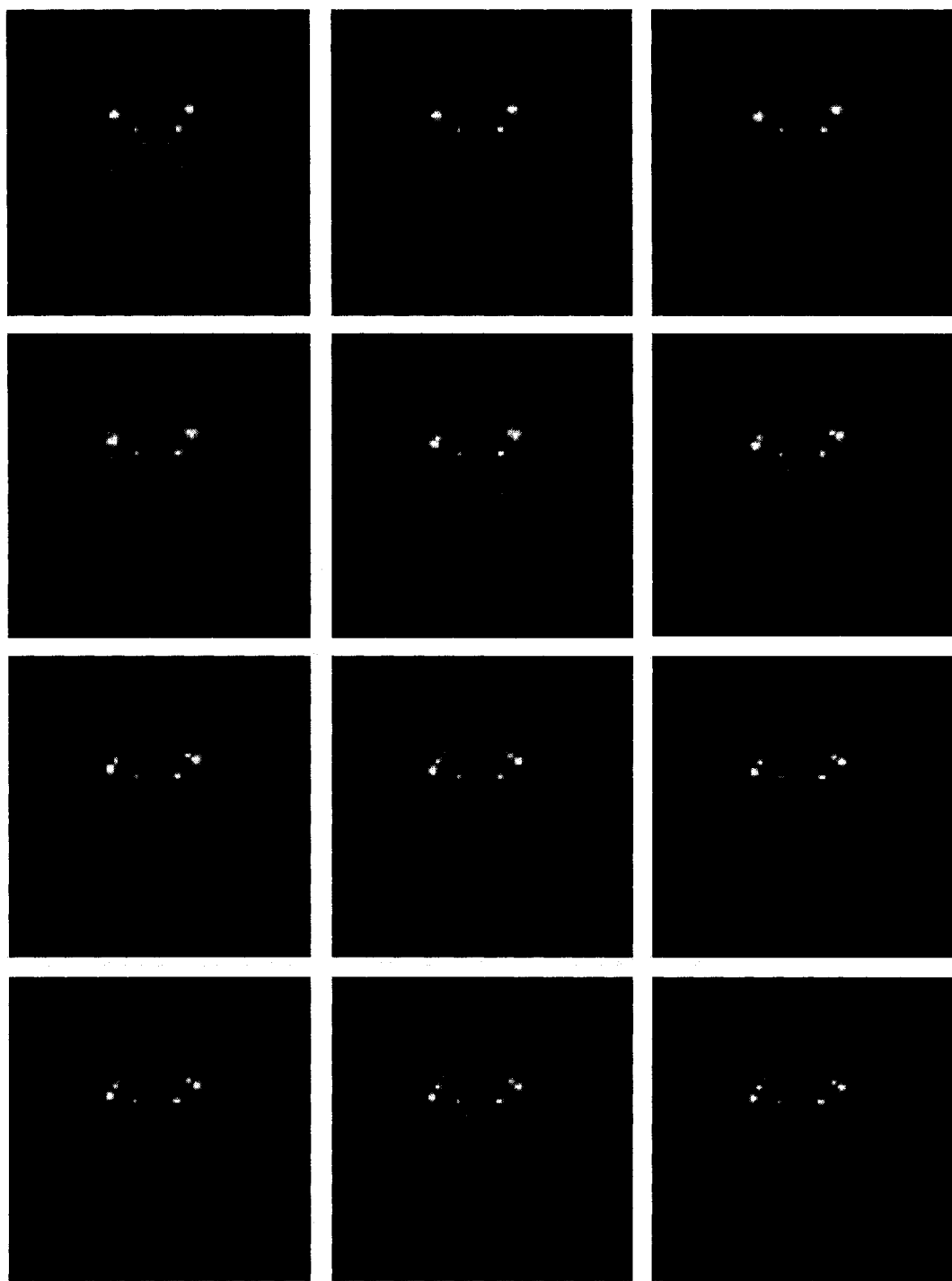


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 94 à 105.

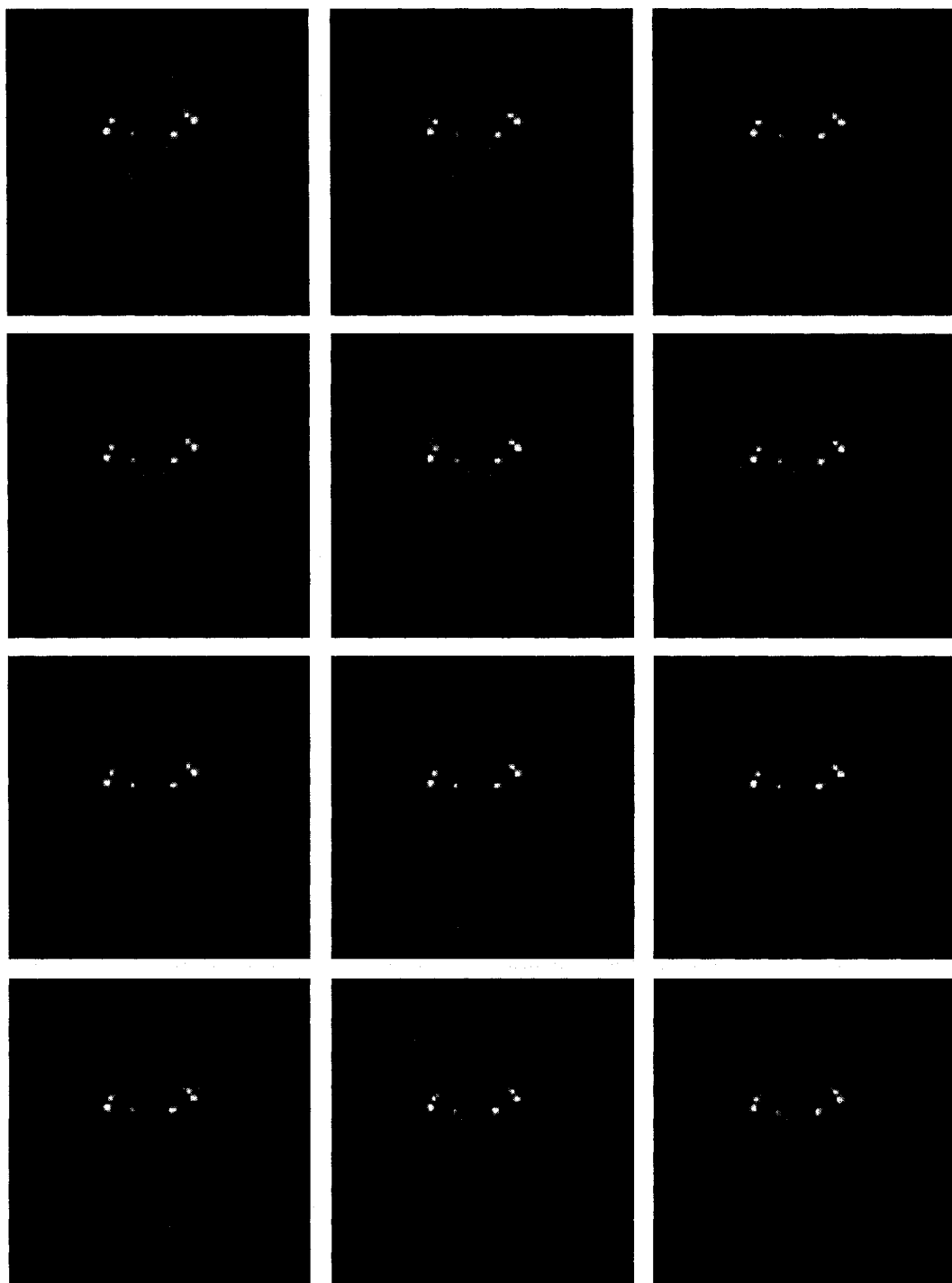


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 106 à 117.

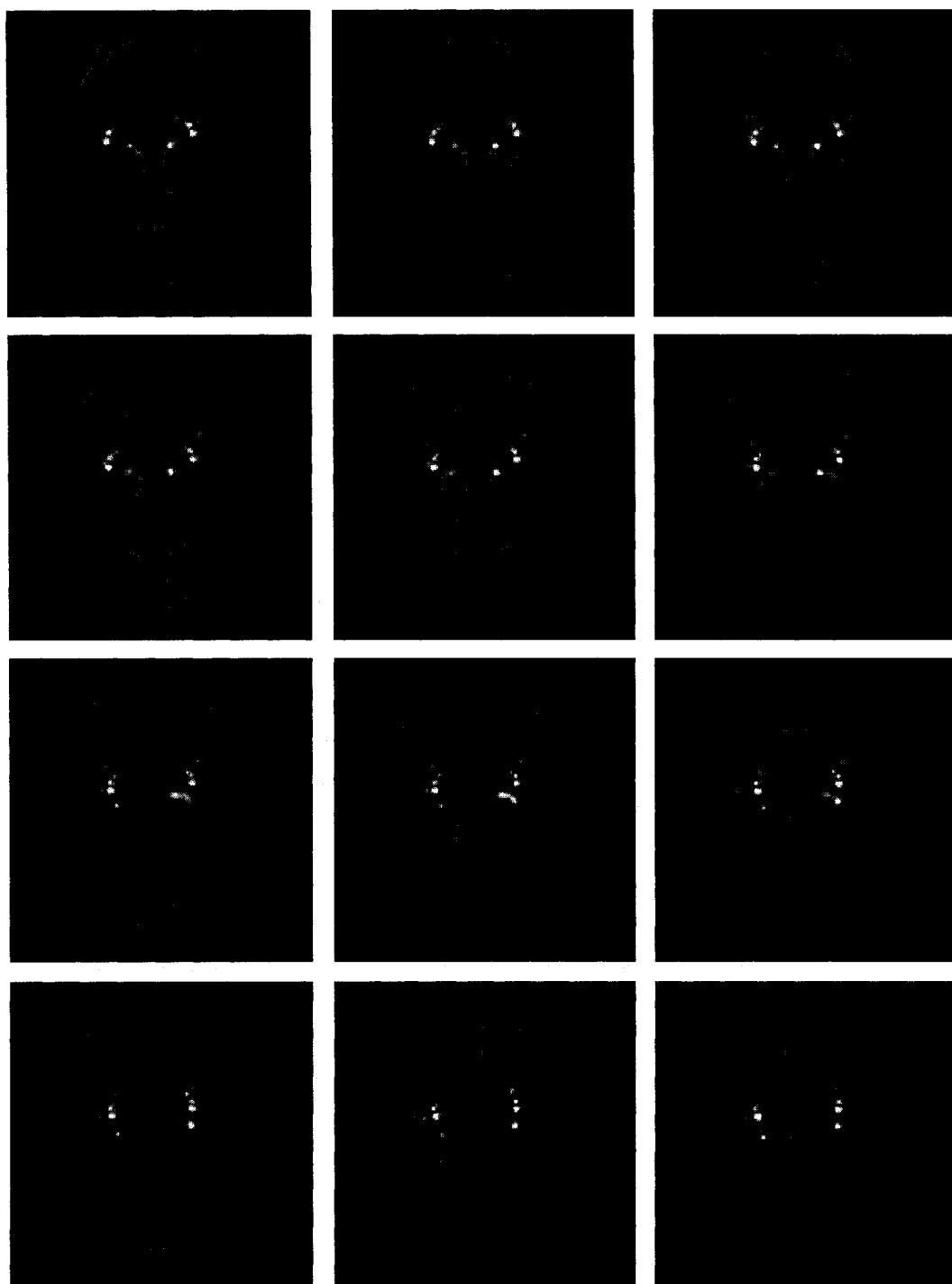
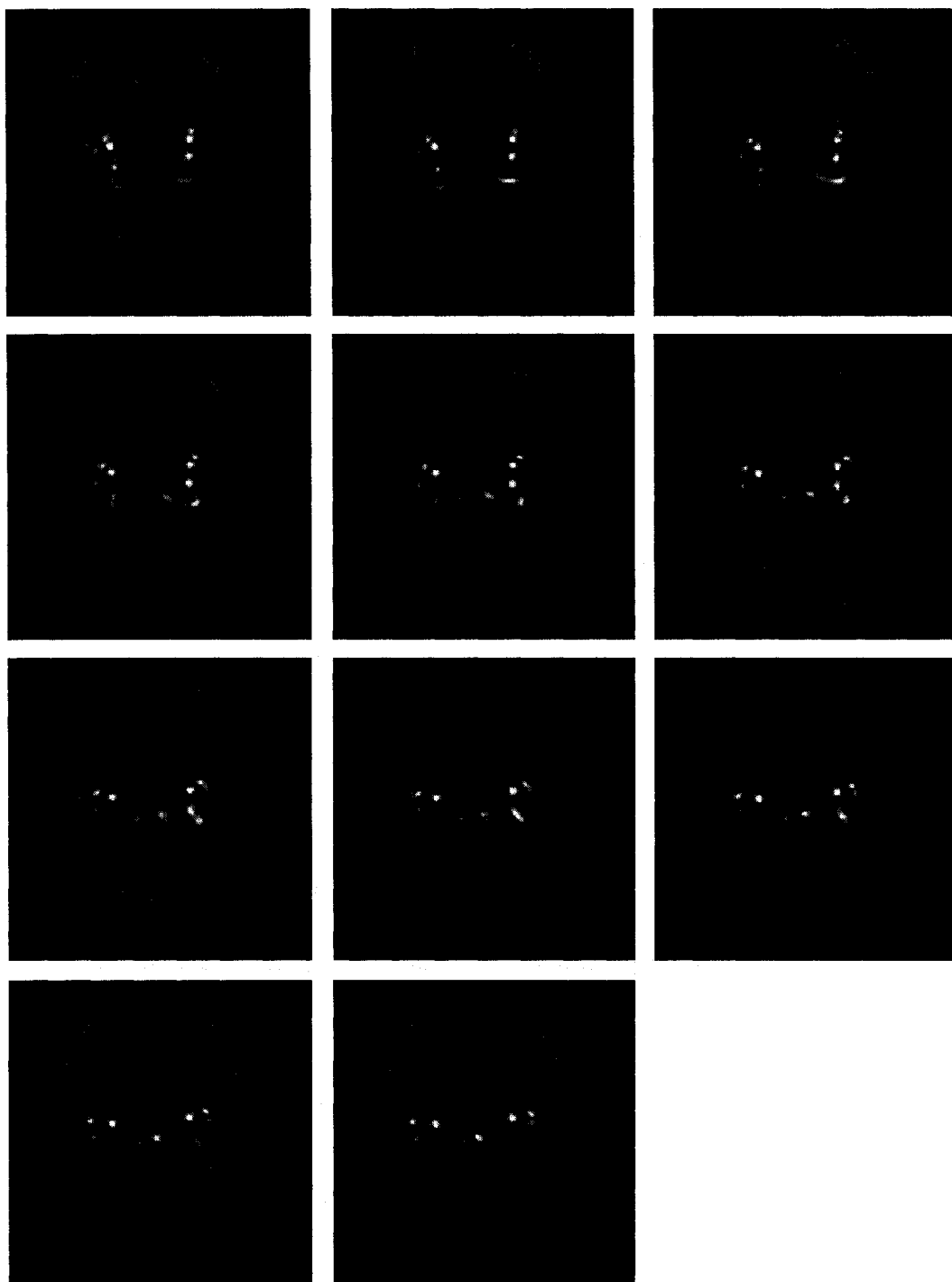


Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 118 à 129.



**Figure A.1 (suite) : Images médicales de la série crosse aortique et carotides. Images 130 à 140.**



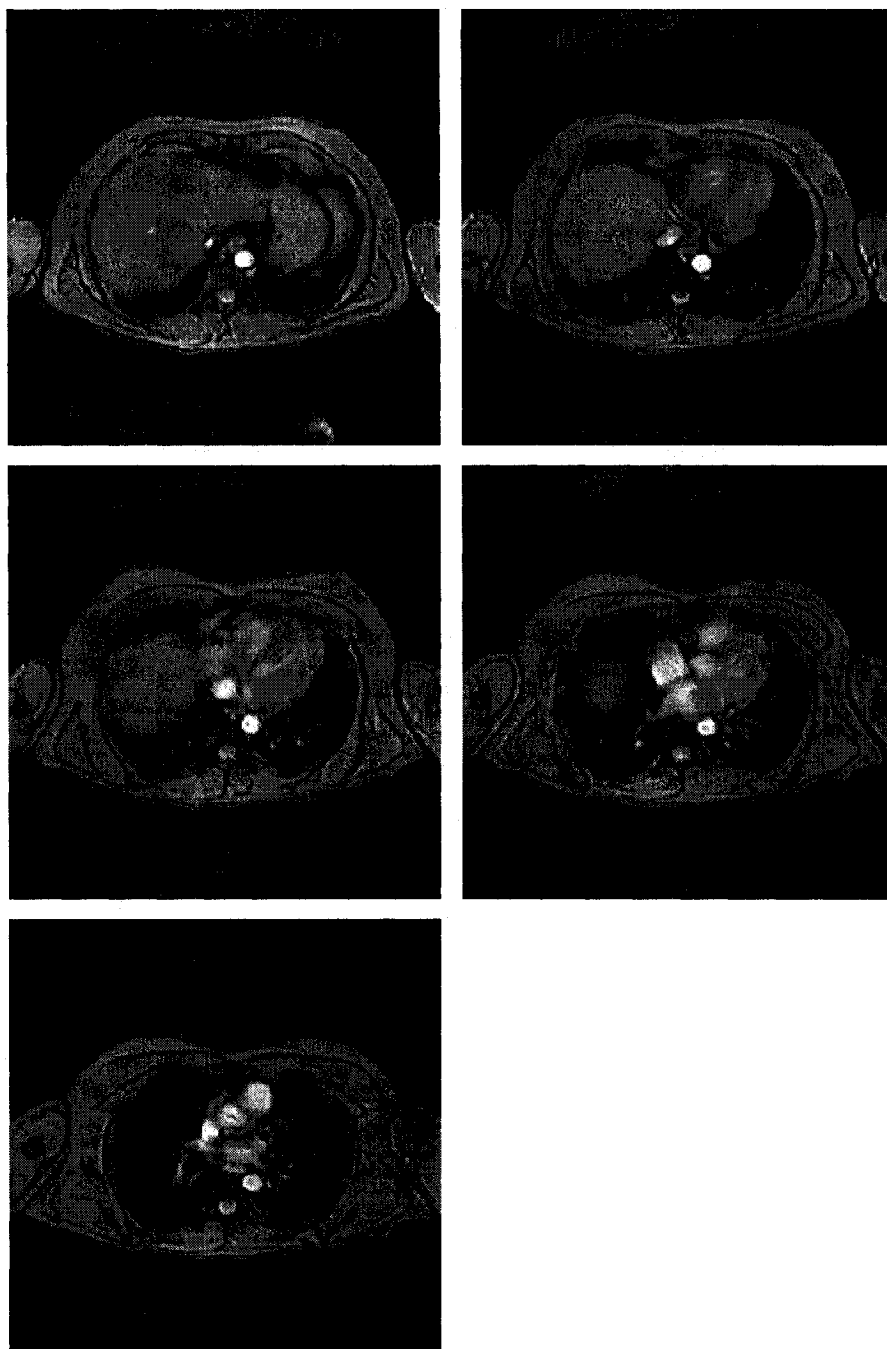


Figure A.2 : Images médicales de la série aorte. Images 1 à 5.

## ANNEXE B : DESCRIPTION DES ALGORITHMES DES FILTRES PROPOSÉS

Cette annexe présente les algorithmes des différents filtres proposés pour le traitement des données à travers les figures B.1 à B.5. Le symbole BX sert à identifier la boucle X, de même que le symbole CY sert à identifier la condition Y. Une ligne de commentaires débute par le symbole `/**`. Ces algorithmes sont implantés dans les fonctions « executer » des classes « TraitementA3D », « TraitementB3D », etc., dont l'interface est illustrée à la figure D.6.

### B.1 Augmentation du contraste

L'algorithme d'augmentation du contraste retenu est illustré à la figure B.1.

```

/** Paramètres d'entrée :
/**      I_initiale = données avant traitement
/**      Seuil_min = seuil minimal
/**      Seuil_max = seuil maximal
/** Paramètres de sortie :
/**      I_finale = données après traitement

B1 : Pour chaque image k des données
  B2 : Pour chaque rangée j des données
    B3 : Pour chaque colonne i des données
      I_finale(i,j,k)  $\leftarrow$  (I_initiale(i,j,k) - Seuil_min) *
      [I_max / (Seuil_max - Seuil_min)]
      Restreindre I_finale(i,j,k) entre 0.0 et 1.0
    Fin B3
  Fin B2
Fin B1

```

**Figure B.1 : Algorithme d'augmentation du contraste.**

Ici,  $I_{\max}$ , qui représente la valeur maximale possible pour un voxel, est égale à 1,0.

## B.2 Filtre de moyenne pondérée

La figure B.2 présente l'algorithme du filtre de moyenne pondérée retenu.

```

/** Paramètres d'entrée :
/**      h = noyau du filtre de moyenne pondérée
/**      I_initiale = données avant traitement
/** Paramètres de sortie :
/**      I_finale = données après traitement

B1 : Pour chaque image k des données
  B2 : Pour chaque rangée j des données
    B3 : Pour chaque colonne i des données
      I_finale(i,j,k) ← 0
      B4 : Pour n entier variant de -1 à +1
        B5 : Pour m entier variant de -1 à +1
          B6 : Pour l entier variant de -1 à +1
            I_finale(i,j,k) ← I_finale(i,j,k) +
              [I_initiale(i+l,j+m,k+n) * h(l,m,n)]
          Fin B6
        Fin B5
      Fin B4
    Fin B3
  Fin B2
Fin B1

```

**Figure B.2 : Algorithme du filtre de moyenne pondérée.**

L'algorithme du filtre de moyenne pondérée est une convolution entre les données et le noyau (de dimensions 3 x 3 x 3 dans le cas présent). Le noyau est représenté ici par  $h$  et  $h(l,m,n)$  symbolise le poids  $W$  correspondant aux indices  $l,m,n$ . La valeur du poids  $W_i$  est affectée aux indices  $(-1,0,0)$  et  $(1,0,0)$ , celle du poids  $W_j$  aux indices  $(0,-1,0)$  et  $(0,1,0)$ , celle du poids  $W_k$  aux indices  $(0,0,-1)$  et  $(0,0,1)$  et celle du poids  $W_{\text{autres}}$  à toutes les autres possibilités, sauf pour la valeur aux indices  $(0,0,0)$  du voxel central qui est mise à zéro. Plusieurs possibilités existent pour traiter les frontières du volume de données. Dans ce projet, nous avons retenu l'option d'ajouter une région avec des valeurs nulles aux alentours des frontières, de façon à pouvoir glisser le filtre sur l'ensemble des données.

### B.3 Filtre médian

L'algorithme du filtre médian qui a été retenu est illustré à la figure B.3.

```

/** Paramètres d'entrée :
/**      I_initiale = données avant traitement
/**      taille du noyau (3 ou 5, ici forcée à 3 pour clarté)
/** Paramètres de sortie :
/**      I_finale = données après traitement

B1 : Pour chaque image k des données
  B2 : Pour chaque rangée j des données
    B3 : Pour chaque colonne i des données
      B4 : Pour n entier variant de -1 à +1
        B5 : Pour m entier variant de -1 à +1
          B6 : Pour l entier variant de -1 à +1
            Ajouter I_initiale(i+l,j+m,k+n) dans le Tableau temporaire
          Fin B6
        Fin B5
      Fin B4
    /** Méthode de tri suggérée : tri rapide
    Tri du Tableau temporaire
    I_finale(i,j,k) ← Tableau temporaire(valeur médiane)
  Fin B3
Fin B2
Fin B1

```

**Figure B.3 : Algorithme du filtre médian.**

Dans cette version de l'algorithme, le tableau temporaire trié contenant les valeurs des données initiales correspondant aux voxels du noyau ( $I_{initiale}(i+l,j+m,k+n)$ ) est recalculé à chaque déplacement de la fenêtre. Il serait possible d'améliorer la rapidité de l'algorithme en tenant compte du fait qu'une partie du travail de tri est déjà effectuée puisqu'il y a un recoupement entre les voxels à trier d'une position de la fenêtre et ceux correspondant à la prochaine position (Nikolaidis et Pitas 2001, p. 21).

## B.4 Opérations morphologiques

La figure B.4 présente l'algorithme des opérations morphologiques de dilatation et d'érosion fonctionnelles. Puisque les deux opérations sont très semblables, il est possible de les regrouper sous la même fonction.

```

/** Paramètres d'entrée :
/**      I_initiale = données avant traitement
/**      type d'opération (dilatation ou érosion)
/**      h = élément structurant
/** Paramètres de sortie :
/**      I_finale = données après traitement

B1 : Pour chaque image k des données
  B2 : Pour chaque rangée j des données
    B3 : Pour chaque colonne i des données
      C1 : Si dilatation
        Valeur  $\leftarrow$  0
      C1 : Sinon (érosion)
        Valeur  $\leftarrow$  1
      Fin C1
      B4 : Pour n entier variant de -1 à +1
        B5 : Pour m entier variant de -1 à +1
          B6 : Pour l entier variant de -1 à +1
            C2 : Si h(l,m,n) vaut 1
              C3 : Si dilatation
                C4 : Si I_initiale(i+l,j+m,k+n) > Valeur
                  Valeur  $\leftarrow$  I_initiale(i+l,j+m,k+n)
                Fin C4
              C3 : Sinon (érosion)
                C5 : Si I_initiale(i+l,j+m,k+n) < Valeur
                  Valeur  $\leftarrow$  I_initiale(i+l,j+m,k+n)
                Fin C5
              Fin C3
            Fin C2
          Fin B6
        Fin B5
      Fin B4
      I_finale(i,j,k)  $\leftarrow$  Valeur
    Fin B3
  Fin B2
Fin B1

```

**Figure B.4 : Algorithme de dilatation et d'érosion fonctionnelles.**

La variable Valeur contient le maximum ou le minimum courant, selon l'opération et h représente l'élément structurant, assimilable à un noyau.

## B.5 Croissance de région

L'algorithme de croissance de région développé est illustré à la figure B.5.

```

/** Paramètres d'entrée :
/**      I_initiale = données avant traitement
/**      Germe = position du voxel de départ
/**      Seuil_min = seuil minimal
/**      Seuil_max = seuil maximal
/** Paramètres de sortie :
/**      I_finale = données après traitement

/** Initialisation pour indiquer qu'aucun voxel n'a été traité
B1 : Pour chaque image k des données
    B2 : Pour chaque rangée j des données
        B3 : Pour chaque colonne i des données
            I_finale(i,j,k) ← -1,0
        Fin B3
    Fin B2
Fin B1

/** Initialisation de la région avec le germe
Région(IndiceCourant) ← Germe
IndiceCourant ← IndiceCourant + 1
I_finale(Germe) ← I_initiale(Germe)

B4 : Tant qu'il y a des voxels à traiter (dans la région)
    /** Récupérer le prochain voxel à traiter (VoxelCourant)
    IndiceCourant ← IndiceCourant - 1
    VoxelCourant ← Région(IndiceCourant)

    /** Vérifier les 6 voisins immédiats de VoxelCourant

    /** Premier voisin en i-1 (VIMoins1)
    C1 : Si fenêtre 3x3x3 centrée sur VIMoins1 comprise dans les limites
    des données
        /** Si le premier voisin n'a pas déjà été traité
        C2 : Si I_finale(VIMoins1) == -1,0
            /** Calculer les valeurs nécessaires pour l'effet régulateur
            NbVoxelsFenêtreOK ← 0
            NbVoxelsPlanOK ← 0
            NbVoxelsVImmédiatsOK ← 0

        B5 : Pour n entier variant de -1 à +1
            B6 : Pour m entier variant de -1 à +1
                B7 : Pour l entier variant de -1 à +1
                    C3 : Si I_initiale(VIMoins1 + (l,m,n)) ≥ Seuil_min
                    ET I_initiale(VIMoins1 + (l,m,n)) ≤ Seuil_max
                        NbVoxelsFenêtreOK ← NbVoxelsFenêtreOK + 1

```

```

C4 : Si l == +1
      NbVoxelsPlanOK ← NbVoxelsPlanOK + 1
Fin C4
C5 : Si l == -1 ET m == 0 ET n == 0
      NbVoxelsVImmédiatsOK ← NbVoxelsVImmédiatsOK + 1
Fin C5
C6 : Si l == +1 ET m == 0 ET n == 0
      NbVoxelsVImmédiatsOK ← NbVoxelsVImmédiatsOK + 1
Fin C6
C7 : Si l == 0 ET m == -1 ET n == 0
      NbVoxelsVImmédiatsOK ← NbVoxelsVImmédiatsOK + 1
Fin C7
C8 : Si l == 0 ET m == +1 ET n == 0
      NbVoxelsVImmédiatsOK ← NbVoxelsVImmédiatsOK + 1
Fin C8
C9 : Si l == 0 ET m == 0 ET n == -1
      NbVoxelsVImmédiatsOK ← NbVoxelsVImmédiatsOK + 1
Fin C9
C10 : Si l == 0 ET m == 0 ET n == +1
      NbVoxelsVImmédiatsOK ← NbVoxelsVImmédiatsOK + 1
Fin C10
Fin C3
Fin B7
Fin B6
Fin B5

/** Si le voxel voisin a la bonne intensité et qu'il répond aux
/** règles de régulation, l'inclure dans la région et conserver
/** son intensité comme valeur finale
/** S'il n'a pas la bonne intensité mais qu'il répond aux règles
/** de régulation, l'inclure dans la région et ramener son
/** intensité dans la bonne plage puis la conserver comme valeur
/** finale
C11 : Si I_initiale(VIMoins1) ≥ Seuil_min
ET I_initiale(VIMoins1) ≤ Seuil_max
      C12 : Si NbVoxelsFenêtreOK > 9 OU NbVoxelsPlanOK > 3
            Région(IndiceCourant) ← VIMoins1
            IndiceCourant ← IndiceCourant + 1
            I_finale(VIMoins1) ← I_initiale(VIMoins1)
Fin C12
C11 : Sinon (intensité pas dans la bonne plage)
      C13 : Si NbVoxelsFenêtreOK ≥ 21
            OU NbVoxelsVImmédiatsOK == 6
            Région(IndiceCourant) ← VIMoins1
            IndiceCourant ← IndiceCourant + 1
            C14 : Si I_initiale(VIMoins1) < Seuil_min
                  I_finale(VIMoins1) ← Seuil_min
Fin C14
            C15 : Si I_initiale(VIMoins1) > Seuil_max
                  I_finale(VIMoins1) ← Seuil_max
Fin C15
Fin C13
Fin C11
Fin C2

```

```

/**/ Traitement spécial pour les voxels voisins aux frontières
/**/ Pas d'effet régulateur : inclure le voxel voisin si son intensité
/**/ est dans la bonne plage seulement
C1 : Sinon, Si VIMoins1 compris dans les limites des données
    /**/ Si le premier voisin n'a pas déjà été traité
    C16 : Si I_finale(VIMoins1) == -1,0
        C17 : Si I_initiale(VIMoins1) ≥ Seuil_min
            ET I_initiale(VIMoins1) ≤ Seuil_max
            Région(IndiceCourant) ← VIMoins1
            IndiceCourant ← IndiceCourant + 1
            I_finale(VIMoins1) ← I_initiale(VIMoins1)
        Fin C17
    Fin C16
Fin C1

/**/ Refaire étapes comprises dans C1 avec les 5 autres voisins immédiats
/**/ (VIPlus1, VJMoins1, VJPlus1, VKMoins1, VKPlus1).

Fin B4

/**/ Mettre l'intensité des voxels hors région à 0,0
B8 : Pour chaque image k des données
    B9 : Pour chaque rangée j des données
        B10 : Pour chaque colonne i des données
            C18 : Si I_finale(i,j,k) == -1,0
                I_finale(i,j,k) ← 0,0
            Fin C18
        Fin B10
    Fin B9
Fin B8

```

**Figure B.5 : Algorithme de croissance de région.**

L'algorithme débute par l'initialisation de la copie locale (représentée ici par  $I_{\text{finale}}$ ) des données à -1,0 pour signifier qu'aucun des voxels n'a été traité pour l'instant. La variable Région est une structure de données qui sert à stocker temporairement les voxels qui font partie de la région et dont les voisins doivent être examinés. Elle fonctionne comme une pile : les voxels en attente de se faire traiter (dont il faut examiner les voisins) sont ajoutés à la fin de la structure, et ils sont retirés par la fin de la structure au moment du traitement. Pour simplifier et rendre l'algorithme plus rapide, au détriment de l'espace mémoire occupé, il est possible d'allouer au départ un très grand tableau de la taille du nombre total de voxels dans les données, puis de garder la trace de la fin de la file (le prochain emplacement libre) avec un indice courant. De cette façon, la condition de la boucle principale B4 n'est plus vérifiée lorsque cet indice courant est



tombé à la valeur 0. Pour chaque voxel faisant partie de la région, ses six voisins immédiats (partageant une face) sont examinés. La variable  $VIMoins1$  représente le voxel immédiatement voisin du voxel courant en  $i-1$ ,  $VIPlus1$  le voxel immédiatement voisin en  $i+1$ ,  $VJMoins1$  le voxel immédiatement voisin en  $j-1$ , etc. Les variables  $NbVoxelsFenêtreOK$ ,  $NbVoxelsPlanOK$  et  $NbVoxelsVImmédiatsOK$  sont utilisées dans l'application de l'effet régulateur de l'algorithme.  $NbVoxelsFenêtreOK$  conserve le nombre de voxels situés dans la fenêtre de taille  $3 \times 3 \times 3$  voxels centrée sur le voisin considéré (par exemple  $VIMoins1$ ) dont l'intensité se trouve dans la bonne plage de valeurs pour faire partie de la région.  $NbVoxelsPlanOK$  contient le nombre de voxels dans le plan de taille  $3 \times 3$  centré sur le voxel courant, perpendiculaire au vecteur entre le voxel immédiatement voisin et le voxel courant, dont l'intensité se trouve dans la bonne plage de valeurs. Dans le cas du voisin immédiat en  $i - 1$ , ce plan serait le plan JK à la coordonnée  $i$ . Enfin,  $NbVoxelsVImmédiatsOK$  représente le nombre de voxels immédiatement voisins du voisin considéré (par exemple les voisins partageant une face avec  $VIMoins1$ ) ayant une intensité se trouvant dans la bonne plage de valeurs.

Dans le cas des voxels voisins situés à la frontière des données, il est impossible de tirer des règles pour assurer un effet régulateur. Ces voxels frontaliers sont inclus à la région seulement si leur intensité se trouve dans la bonne plage, comme dans l'algorithme de croissance de région classique.

## ANNEXE C : FORMAT DES FICHIERS UTILISÉS

Cette annexe présente les divers formats de fichier utilisés dans le projet : le format « VOL », le format « TRM », le format « VU », le format « YAMS ».

### C.1 Format « VOL »

La figure C.1 illustre le format du fichier « VOL », qui contient les données sources utilisées dans le projet. Ce format a été défini pour nos besoins spécifiques.

```
// VOL 1.0
NombreDeDimensions
[Dt] Dz Dy Dx
[N1] Nk Nj Ni

voxel_1111 voxel_1112 ... voxel_111Ni } 1° dim.
voxel_1121 voxel_1122 ... voxel_112Ni } 2° dim.
...
voxel_11Nj1 voxel_11Nj2 ... voxel_11NjNi

voxel_1211 voxel_1212 ... voxel_121Ni
voxel_1221 voxel_1222 ... voxel_122Ni
...
voxel_12Nj1 voxel_12Nj2 ... voxel_12NjNi

voxel_1Nk11 voxel_1Nk12 ... voxel_1Nk1Ni
voxel_1Nk21 voxel_1Nk22 ... voxel_1Nk2Ni
...
voxel_1NkNj1 voxel_1NkNj2 ... voxel_1NkNjNi

voxel_2111 voxel_2112 ... voxel_211Ni
voxel_2121 voxel_2122 ... voxel_212Ni
...
voxel_21Nj1 voxel_21Nj2 ... voxel_21NjNi

voxel_2211 voxel_2212 ... voxel_221Ni
voxel_2221 voxel_2222 ... voxel_222Ni
...
voxel_22Nj1 voxel_22Nj2 ... voxel_22NjNi

voxel_2Nk11 voxel_2Nk12 ... voxel_2Nk1Ni
```

```

voxel_2Nk21 voxel_2Nk22 ... voxel_2Nk2Ni
...
voxel_2NkNj1 voxel_2NkNj2 ... voxel_2NkNjNi
...

voxel_Nl111 voxel_Nl112 ... voxel_Nl11Ni
voxel_Nl121 voxel_Nl122 ... voxel_Nl12Ni
...
voxel_Nl1Nj1 voxel_Nl1Nj2 ... voxel_Nl1NjNi

voxel_Nl211 voxel_Nl212 ... voxel_Nl21Ni
voxel_Nl221 voxel_Nl222 ... voxel_Nl22Ni
...
voxel_Nl2Nj1 voxel_Nl2Nj2 ... voxel_Nl2NjNi

voxel_NlNk11 voxel_NlNk12 ... voxel_NlNk1Ni
voxel_NlNk21 voxel_NlNk22 ... voxel_NlNk2Ni
...
voxel_NlNkNj1 voxel_NlNkNj2 ... voxel_NlNkNjNi

```

**Figure C.1 : Format de fichier « VOL ».** NombreDeDimensions est un entier prenant la valeur 3 ou 4; D<sub>t</sub>, D<sub>z</sub>, D<sub>y</sub> et D<sub>x</sub> sont nombres décimaux positifs; N<sub>l</sub>, N<sub>k</sub>, N<sub>j</sub> et N<sub>i</sub> sont des nombres entiers positifs non nuls; voxel\_N<sub>l</sub>N<sub>k</sub>N<sub>j</sub>N<sub>i</sub> et autres sont des nombres décimaux entre 0,0 et 1,0.

Le fichier est de type texte. Les quatre premières lignes constituent l'entête du fichier. La première ligne sert à identifier le format « VOL », la seconde indique le nombre de dimensions, la troisième donne l'espacement entre les centres de deux voxels voisins dans les différentes directions (D<sub>t</sub> ne sera présent que si les données sont en 4D), et la quatrième précise la taille des données (nombre de voxels) dans chaque direction (N<sub>l</sub> ne sera présent que si les données sont en 4D). Les valeurs des voxels sont ensuite disposées de manière à faire varier la direction i le plus rapidement, suivie de la direction j, la direction k et la direction l (seulement en 4D). Dans le cas d'une série d'images médicales 4D, une rangée d'une image est représentée par une ligne de valeurs, une image est représentée par un groupe de lignes sans espacement, les images sont séparées entre elles par une ligne d'espacement et les séries à divers temps sont séparées par deux lignes d'espacement. Les valeurs des voxels (voxel\_N<sub>l</sub>N<sub>k</sub>N<sub>j</sub>N<sub>i</sub>) sont des nombres décimaux entre 0,0 et 1,0.

## C.2 Format « TRM »

La figure C.2 illustre le format très simple du fichier « TRM », qui contient les commandes des traitements à effectuer sur les données sources avec leurs paramètres, exécutées dans leur ordre de présentation. Ce format, utilisé en version console, a été défini pour les besoins spécifiques du projet. Il est ouvert, c'est-à-dire qu'il peut être modifié pour accommoder de nouvelles commandes.

```
NomDeLaCommande1 Parametre1 Parametre2 ...  
NomDeLaCommande2 Parametre1 Parametre2 ...  
...  
NomDeLaCommandeM Parametre1 Parametre2 ...
```

**Figure C.2 : Format de fichier « TRM ».** **NomDeLaCommandeM** est une chaîne de caractère; **Parametre1** et suivants peuvent être des nombres ou des chaînes de caractères.

Le fichier est de type texte. Chaque ligne représente un traitement différent à effectuer, l'ensemble des traitements étant proposé par les M commandes. Le nombre de paramètres dépend de la commande en question. Les commandes possibles (actuelles), de même que leurs paramètres sont donnés au tableau C.1.

Tableau C.1 : Commandes et paramètres pour le format de fichier « TRM ».

Traitement	Nom de la commande	Nombre de paramètres	Liste des paramètres
Augmentation du contraste	CONT	2	- seuil minimal - seuil maximal (nombres décimaux entre 0 et 1)
Filtre de moyenne pondérée	MOY	5	- poids immédiat en i - poids immédiat en j - poids immédiat en k - poids immédiat en l - poids des autres (nombre entiers positifs)
Filtre médian	MED	1	- taille de la fenêtre du filtre (3 ou 5)
Opération morphologique	OP_MORPH	2	- type d'opération (« DIL » ou « ERO », pour dilatation ou érosion, respectivement) - type d'élément structurant (« 3DCube2 » ou « 3DCube3 » ou « 3DCroix3 », pour un élément 2x2x2 avec des '1' partout, un élément 3x3x3 avec des '1' partout ou un élément 3x3x3 avec 5 '1' disposés en croix, respectivement)
Croissance de région	REG	2	- seuil minimal - seuil maximal (nombres décimaux entre 0 et 1)

### C.3 Format « VU »

La figure C.3 montre le format du fichier « VU » tel qu'il est produit par notre programme. Le format complet supporte de nombreux autres champs et fonctionnalités (inVisu 2005).

```

CHAMP NomListeDesSommets () =
{
CoordXi      CoordYi      CoordZi
...
CoordXn      CoordYn      CoordZn
};

CHAMP<int> NomListeDesElements () =
{
NoSommet1i NoSommet2i NoSommet3i
...
NoSommet1n NoSommet2n NoSommet3n
};

MAILLAGE NomMaillage () =
{
ZONE NomListeDesElements ( LagrTrian03 , NomListeDesSommets%3 ,
NomListeDesElements);
};

```

**Figure C.3 : Format de fichier « VU ».** Les mots débutant par « Nom » sont des chaînes de caractères; les mots débutant par « Coord » sont des nombres décimaux; les mots débutant par « NoSommet » sont des entiers positifs non nuls.

Les mots en caractères majuscules correspondent à des mots clés du format. Le fichier, divisé en trois sections, débute par la liste de tous les sommets de la surface. Chaque ligne donne les coordonnées d'un sommet dont le nombre est variable et dépend du type d'élément. À la deuxième section se trouve la liste des éléments du maillage, dans ce cas des triangles. Chaque ligne donne, pour les sommets d'un triangle, les numéros de référence à trois sommets de la liste des sommets. Enfin, le fichier se termine par la définition de la structure du maillage. L'utilisation de triangles est spécifiée par le mot clé « LagrTrian03 », et le nombre de coordonnées par le « %3 » après le nom de la liste des sommets.

## C.4 Format « YAMS »

La figure C.4 illustre le format du fichier « YAMS » tel qu'il est produit par notre programme. Le format complet (appelé « mesh ») supporte de nombreux autres champs (Frey 2001). Nous nous sommes limités aux champs obligatoires.

```
# Commentaire
MeshVersionFormatted    1
Dimension    3

# Ensemble des sommets
Vertices
NombreDesSommetsDifférents
CoordXi    CoordYi    CoordZi    0
...
CoordXn    CoordYn    CoordZn    0

# Ensemble des arêtes
Edges
NombreDArêtes
NoSommet1i NoSommet2i    0
...
NoSommet1n NoSommet2n    0

# Ensemble des éléments
Triangles
NombreDEléments
NoSommet1i NoSommet2i    NoSommet3i    0
...
NoSommet1n NoSommet2n    NoSommet3n    0

# Ensemble des normales
Normals
NombreDeVecteursNormauxDifférents
CoordXi    CoordYi    CoordZi    0
...
CoordXn    CoordYn    CoordZn    0

# Ensemble des normales aux sommets
NormalAtVertices
NombreDeNormales
NoSommeti    NoVecteurNormali
...
NoSommetn    NoVecteurNormaln

END
```

**Figure C.4 : Format de fichier « YAMS ».** Les mots débutant par « Nombre » sont des entiers positifs; les mots débutant par « Coord » sont des nombres décimaux; les mots débutant par « No » sont des entiers positifs non nuls.

Une ligne de commentaire débute avec le symbole #. Les deux premières lignes constituent l'entête et sont obligatoires. Le fichier comprend ensuite une description d'un maillage tridimensionnel de triangles. Cette description, divisée en cinq parties, débute par la liste de tous les sommets, spécifiée par le mot clé « Vertices », suivi du nombre total de sommets. Les coordonnées d'un sommet (nombres décimaux) se retrouve sur la même ligne. À la suite se trouve la liste des arêtes, spécifiée par le mot clé « Edges », suivi du nombre d'arêtes. Chaque ligne donne, pour les sommets d'une arête, les numéros de référence à deux sommets de la liste des sommets. Vient ensuite la liste des éléments, spécifiée par le mot clé « Triangles », suivi du nombre d'éléments. Chaque ligne donne, pour les sommets d'un triangle, les numéros de référence à trois sommets de la liste des sommets. Dans la quatrième partie se trouve la liste des vecteurs normaux, semblable à la liste des sommets. Elle est spécifiée par le mot clé « Normals », suivi du nombre total de vecteurs normaux. Les coordonnées d'un vecteur normal se retrouve sur la même ligne. Enfin, en dernière partie, l'association entre les vecteurs normaux et les sommets est décrite par la liste des normales aux sommets, spécifiée par le mot clé « NormalAtVertices », suivi du nombre de normales. Chaque ligne fournit le numéro de référence à un vecteur normal de la liste des vecteurs normaux pour chaque sommet de la liste des sommets. Le fichier se termine par le mot clé « END ».



## ANNEXE D : DIAGRAMMES DES SOUS-SYSTÈMES

Cette annexe regroupe les diagrammes illustrant les principales méthodes et attributs des 11 sous-systèmes constituant le logiciel développé. Certains attributs et fonctions membres ont été omis afin d'alléger les diagrammes (constructeurs, destructeurs, fonctions d'accès et autres). Les flèches pleines indiquent les relations d'agrégation et les flèches en pointillés les relations d'association.

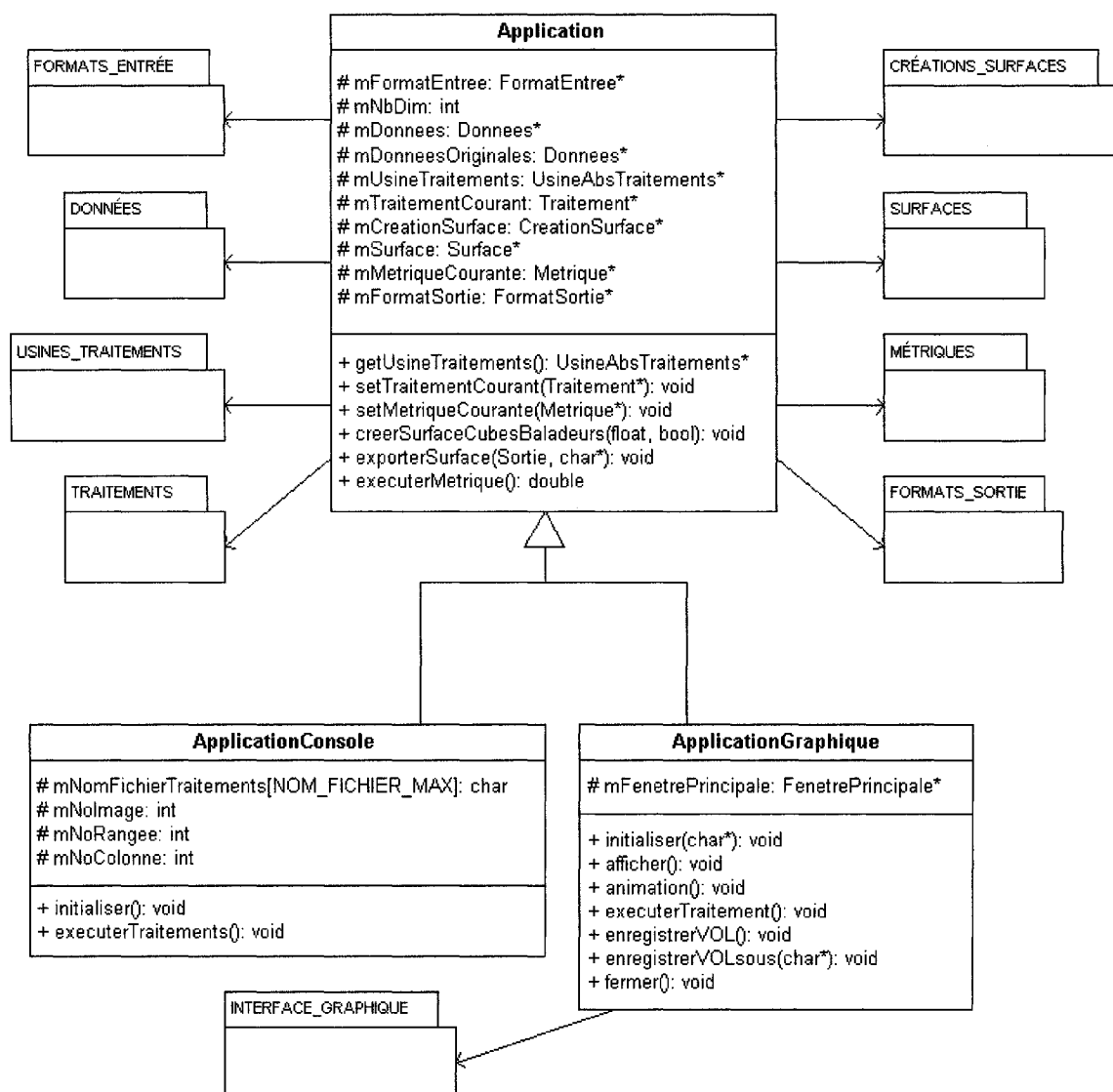


Figure D.1 : Diagramme du sous-système « Application ».

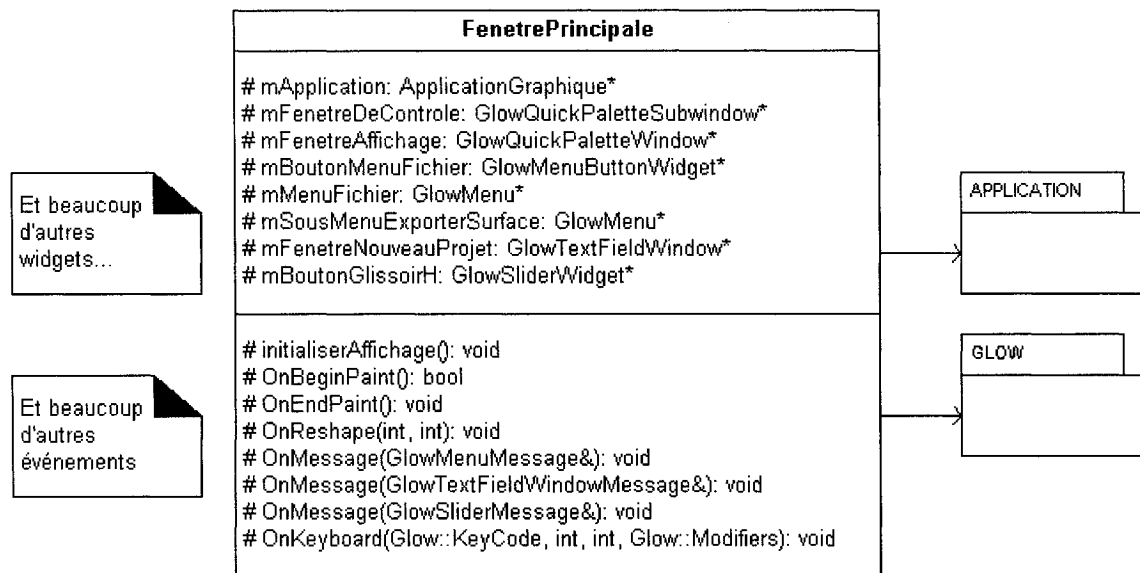


Figure D.2 : Diagramme du sous-système « Interface Graphique ».

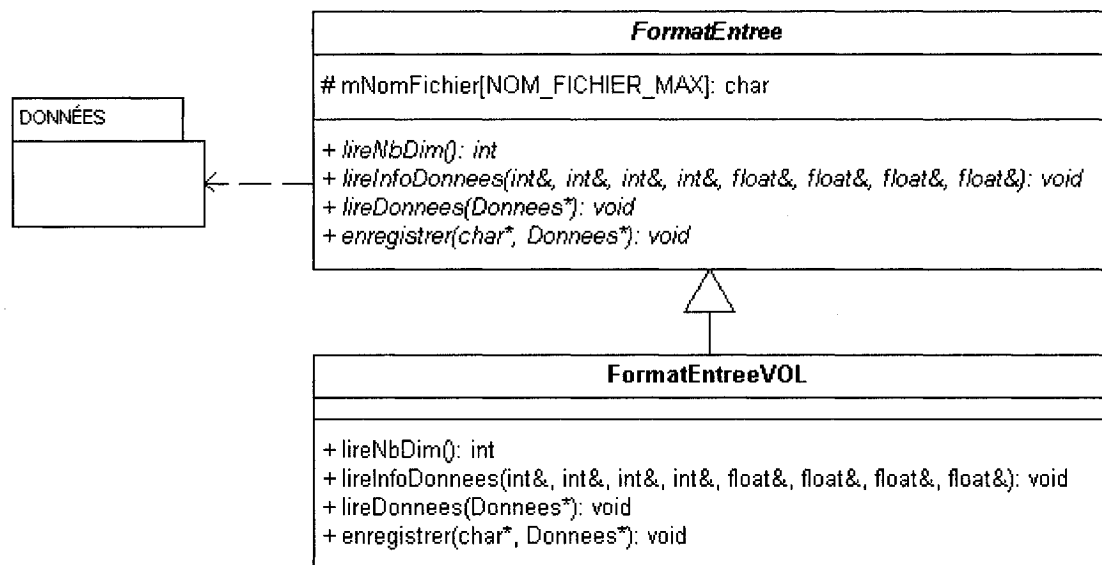


Figure D.3 : Diagramme du sous-système « Formats Entrée ». Présence du patron de conception « stratégie ».

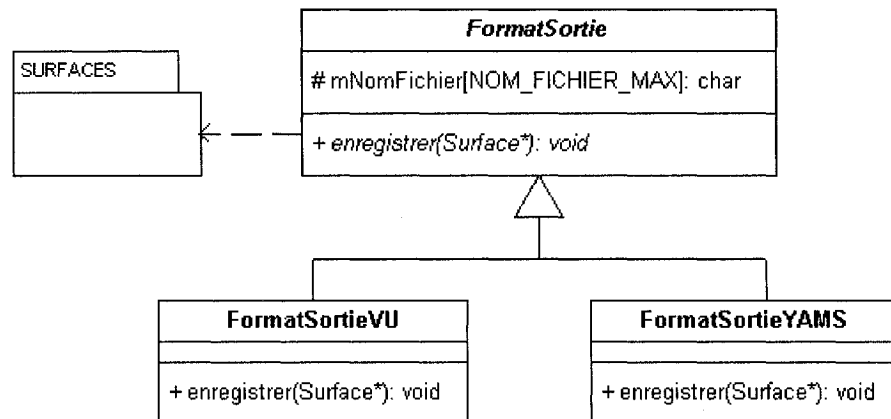


Figure D.4 : Diagramme du sous-système « Formats Sortie ». Présence du patron de conception « stratégie ».

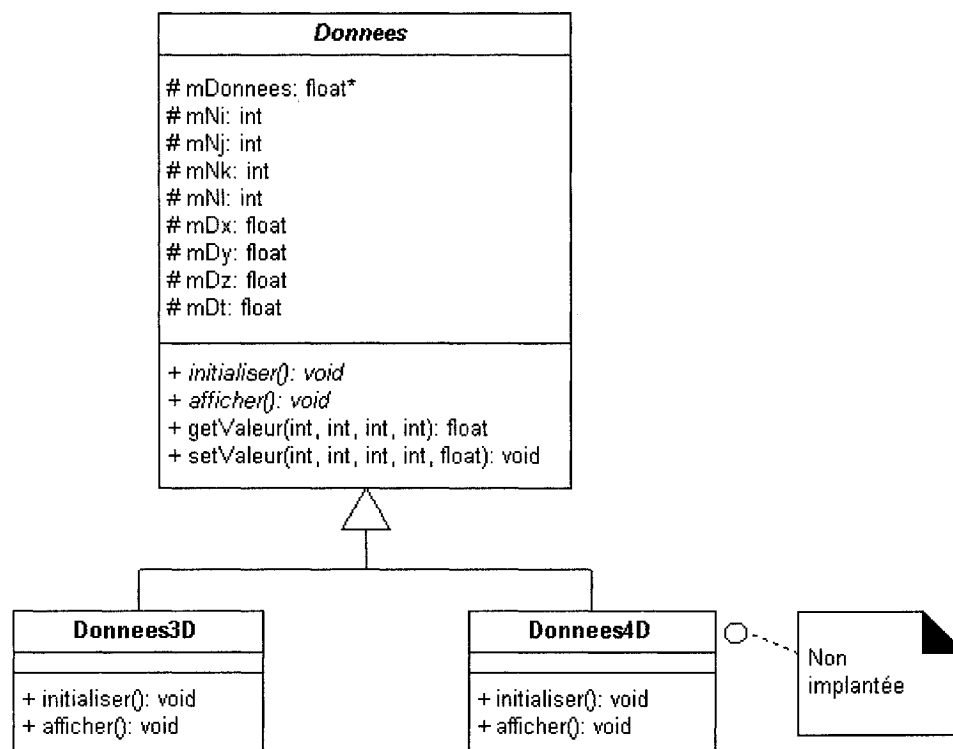
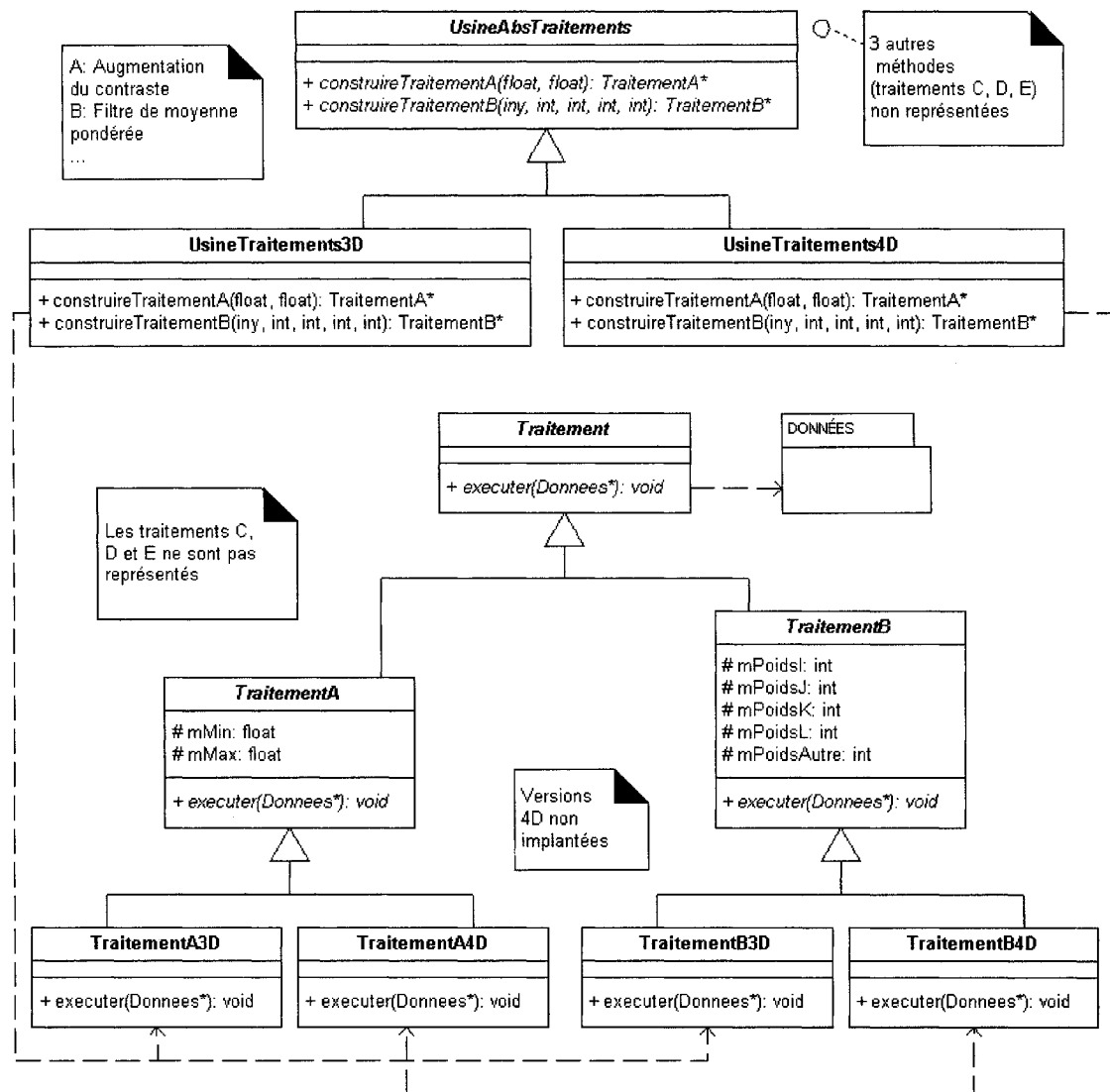
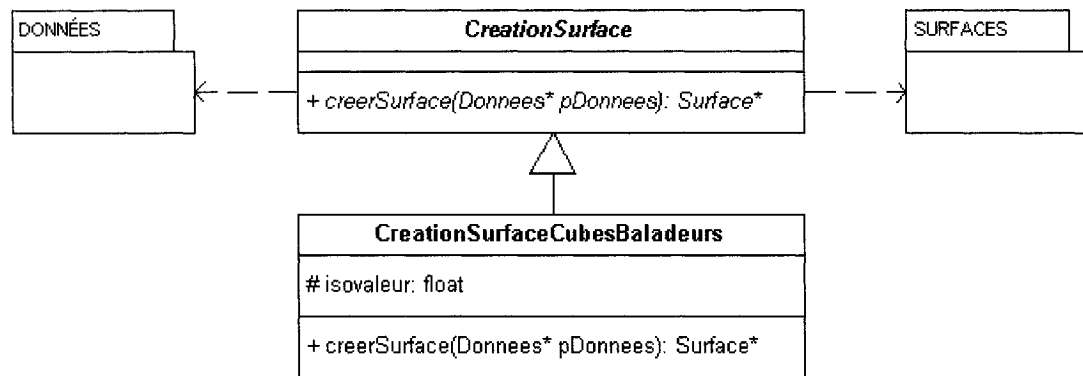


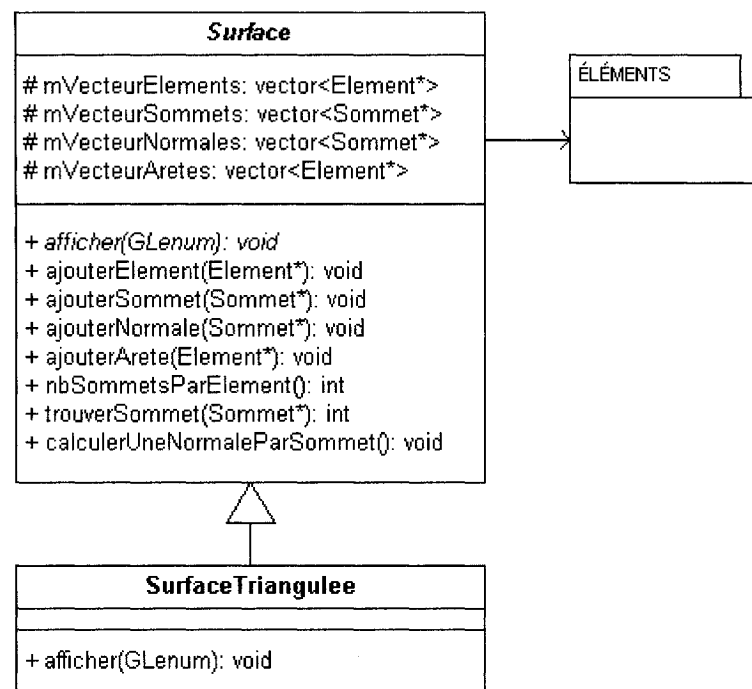
Figure D.5 : Diagramme du sous-système « Données ».



**Figure D.6 : Diagramme des sous-systèmes « Usines Traitements » et « Traitements ». Présence des patrons de conception « usine abstraite » et « usine ».**



**Figure D.7 : Diagramme du sous-système « Créations Surfaces ». Présence du patron de conception « stratégie ».**



**Figure D.8 : Diagramme du sous-système « Surfaces ».**

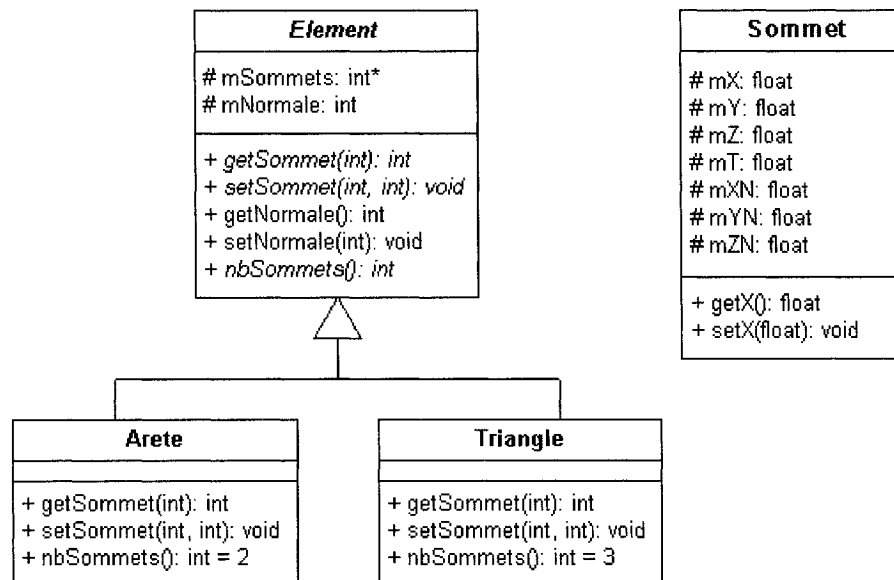


Figure D.9 : Diagramme du sous-système « Éléments ».

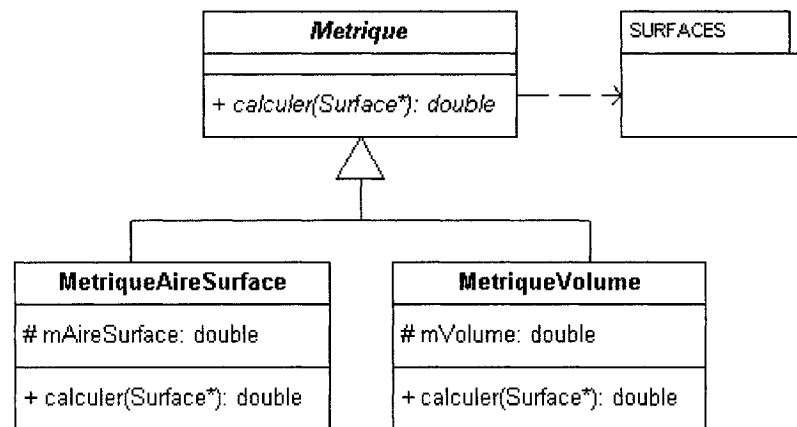


Figure D.10 : Diagramme du sous-système « Métriques ». Présence du patron de conception « stratégie ».

## ANNEXE E : FONCTIONNEMENT ET EXÉCUTION D'ASSIMMO3D

Cette annexe rassemble diverses informations concernant le fonctionnement et l'exécution du logiciel développé. La figure E.1 indique la syntaxe à suivre lors du lancement du programme en mode console. Le symbole « \ » indique que la commande se poursuit à la ligne suivante.

### SYNTAXE

```
ASSIMMO3D.exe -p noImage noRangée noColonne \  
[ -e fichierEntrée.vol ] [ -c fichierCommandes.trm ] \  
[ -t { VU | YAMS } ] [ -s fichierSortie ]
```

-p Spécifier un point situé dans l'organe sur l'image  
« noImage », la rangée « noRangée » et la colonne  
« noColonne ».

Attention, la numérotation commence à zéro (0).

### OPTIONS:

- e Spécifier le nom du fichier d'entrée de données (type VOL).  
Par défaut: « donnees.vol ».
- c Spécifier le nom du fichier contenant les traitements à  
appliquer sur les données (type TRM).  
Par défaut: « commandes.trm ».
- t Spécifier le type de fichier de sortie:  
VU: compatible avec le logiciel VU  
YAMS: compatible avec le logiciel YAMS  
Par défaut: « VU »
- s Spécifier le nom du fichier de sortie.  
Par défaut: « sortie.{vu | mesh} »

**Figure E.1 : Syntaxe à suivre lors du lancement d'ASSIMMO3D en mode console.**

À la figure E.2 se trouve un diagramme de séquence UML, illustrant les interactions entre les objets en mode console. Certains objets ont été omis (éléments polygonaux, métriques, etc.) afin d'alléger le contenu du diagramme. Les messages pour « ApplicationConsole » provenant de la gauche du diagramme sont envoyés par le programme principal (*main*).

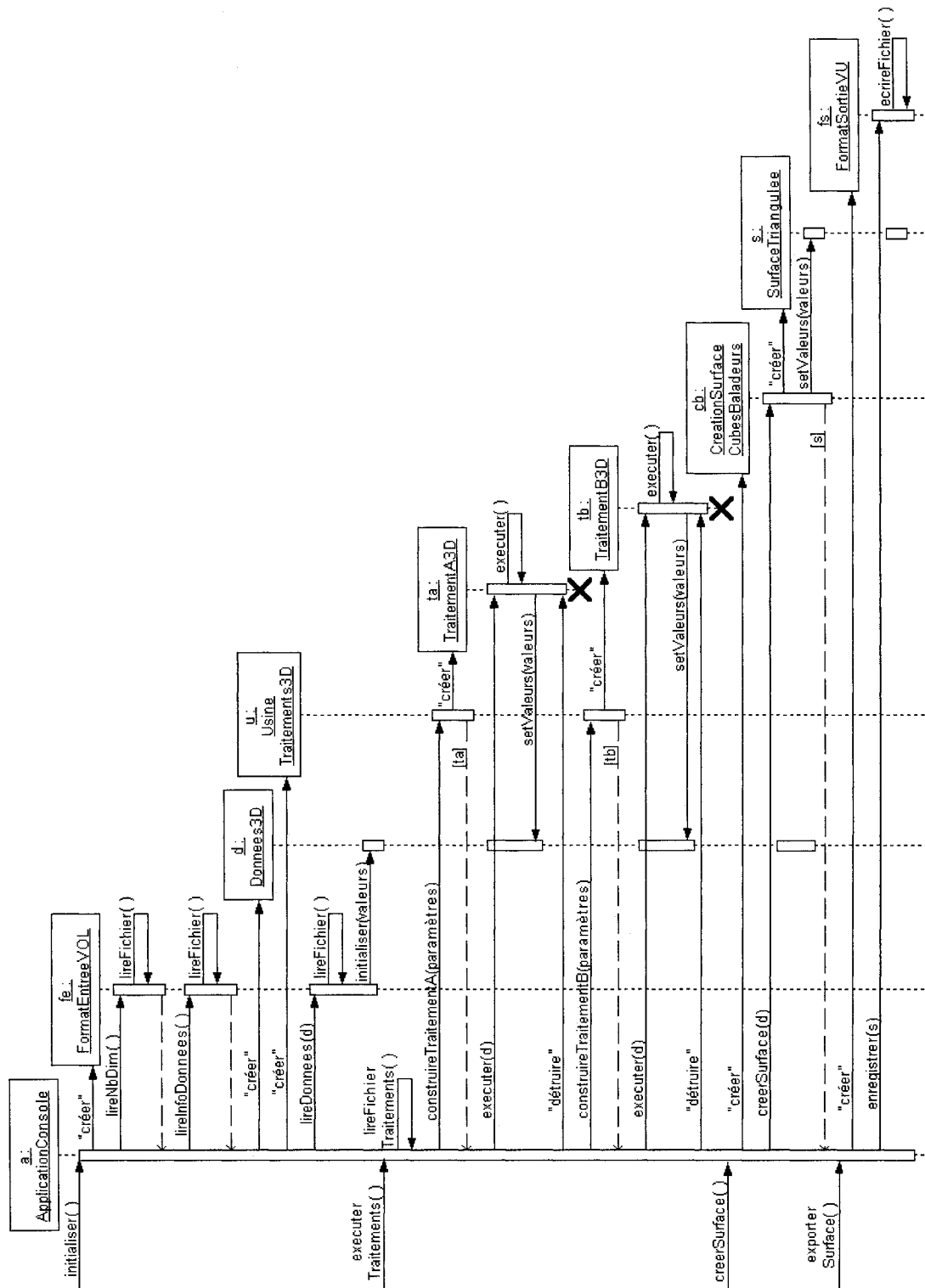


Figure E.2 : Diagramme de séquence UML pour le mode console.



La figure E.3 montre l'interface usager graphique de la fenêtre de contrôle pour l'affichage des données et des surfaces en mode interactif.

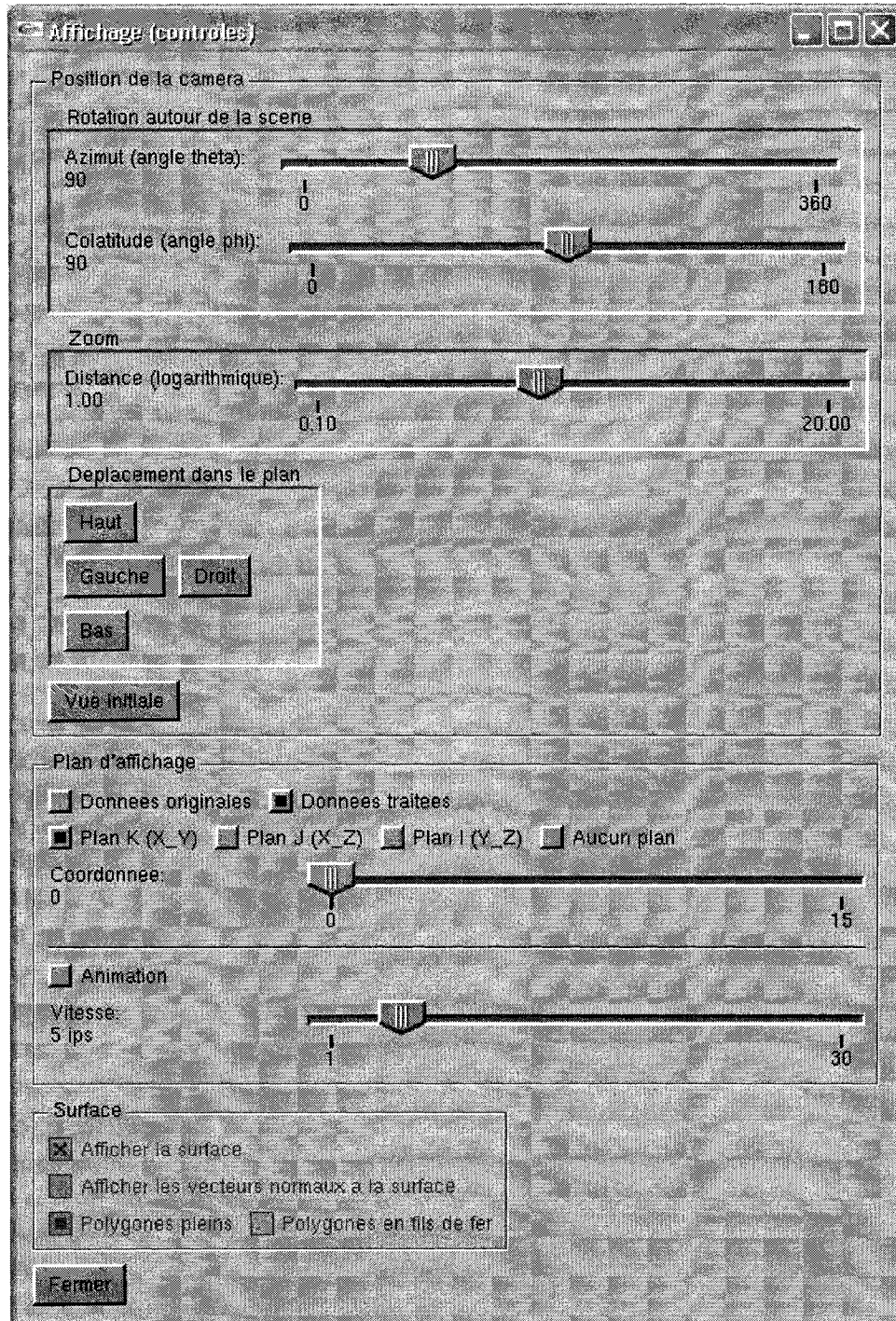
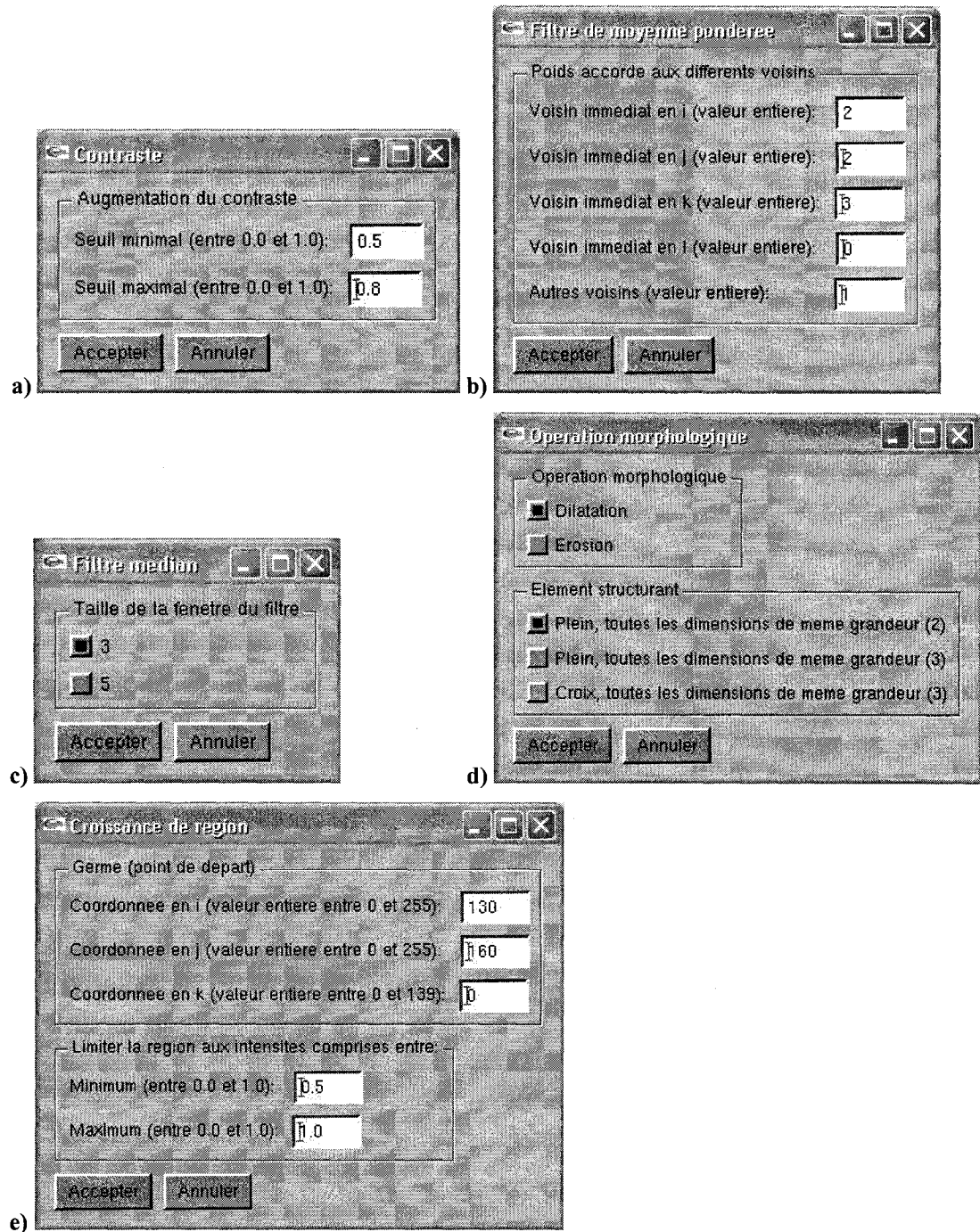


Figure E.3 : Fenêtre de contrôle de l'affichage d'ASSIMMO3D en mode interactif.

La figure E.4 présente l'interface usager graphique pour le choix des paramètres des traitements disponibles.



**Figure E.4 : Interface usager graphique pour le choix des paramètres pour les traitements. En a), augmentation du contraste; en b), filtre de moyenne pondérée; en c), filtre médian; en d), opération morphologique; en e), croissance de région.**